



TESIS – IF185401

**ANALISIS DOKUMEN SKPL (SPESIFIKASI
KEBUTUHAN PERANGKAT LUNAK)
MENGUNAKAN KLASTERISASI SPECTRAL**

PATRICIA GERTRUDIS MANEK

NRP. 05111650010043

DOSEN PEMBIMBING

Daniel Oranova Siahaan, S.Kom, M.Sc, PD.ENG.

PROGRAM MAGISTER

BIDANG KEAHLIAN REKAYASA PERANGKAT LUNAK

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2019

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M. Kom)
di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

PATRICIA GERTRUDIS MANEK
Nrp. 05111650010043

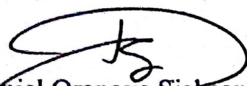
Dengan judul:

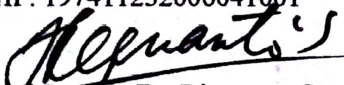
Analisis Dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak)
Menggunakan Klasterisasi *Spectral*

Tanggal Ujian : 26-10-2018


Periode Wisuda : 2019 Ganjil

Disetujui oleh:


Daniel Oranova Siahaan, S.Kom, M.Sc, PD.Eng (Pembimbing)
NIP. 197411232006041001


Prof. Ir. Drs. Ec. Riyanarto Sarno, M.Sc., Ph.D (Penguji 1)
NIP. 195908031986011001


Dr. Ir. Siti Rochimah, M.T (Penguji 2)
NIP. 196810021994032001


Dr. Eng. Radityo Anggoro, S.Kom., M.Sc (Penguji 3)
NIP. 1984101620081210002

Dekan Fakultas Teknologi Informasi dan Komunikasi,


Dr. Agus Zainal Arifin, S.Kom, M.Kom
NIP. 19720809 199512 1 001

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tesis saya dengan judul:

ANALISIS DOKUMEN SKPL (SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK) MENGGUNAKAN KLASTERISASI *SPECTRAL*

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 21 Januari 2019

Patricia G. Manek

[Halaman ini sengaja dikosongkan]

ANALISIS DOKUMEN SKPL (SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK) MENGGUNAKAN KLASTERISASI SPECTRAL

Nama : Patricia Gertrudis Manek
NRP : 5116201043
Pembimbing : Daniel Oranova Siahaan, S.Kom, M.Sc,PD.Eng.

ABSTRAK

Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan spesifikasi fungsional (kapabilitas/fungsi) dan non-fungsional (batasan, karakteristik, kualitas, dan properti) dari perangkat lunak yang akan dikembangkan berdasarkan kondisi atau kemampuan yang harus dimiliki perangkat lunak ataupun yang dibutuhkan oleh pengguna. Bagi pihak pengembang, dokumen SKPL digunakan sebagai acuan dalam setiap tahapan pengembangan. Dalam proses pembuatan dokumen SKPL seringkali terdapat kesalahan. Salah satu kesalahan tersebut yang dirumuskan oleh Meyer sebagai *The Seven Sins of Specifier* adalah *noise* atau derau. Derau di dalam SKPL dapat muncul pada proses spesifikasi kebutuhan pengembangan perangkat lunak ketika menyertakan suatu informasi yang tidak relevan terhadap kebutuhan pengguna.

Penelitian ini bertujuan mendeteksi derau di dalam daftar pernyataan kebutuhan dari suatu dokumen SKPL. Pendeteksian derau menggunakan metode *spectral clustering*. Setiap pernyataan kebutuhan dirupakan sebagai suatu vector yang terdiri dari frekuensi kemunculan kata-kata unik dalam spesifikasi kebutuhan tersebut. Dengan metode *spectral clustering*, pernyataan kebutuhan yang mengandung derau dipisahkan dari kelompok kebutuhan yang bukan derau. Klaster yang memiliki rata-rata jarak individu terhadap pusat klaster yang paling jauh merupakan klaster pernyataan kebutuhan yang mengandung derau.

Hasil uji coba menunjukkan bahwa metode yang dikembangkan memiliki tingkat sensitivitas yang baik, yaitu 1,0, baik untuk data asli maupun data sintetis. Akan tetapi, hasil uji coba juga menunjukkan bahwa metode yang dikembangkan memiliki tingkat Spesivitas yang kurang baik, khususnya untuk data asli, yaitu 0,19, dibanding dengan data sintetis, yaitu 0,83.

Kata kunci: Derau, Deteksi Derau, Klasterisasi Spektral, Spesifikasi Kebutuhan

[Halaman ini sengaja dikosongkan]

DOCUMENT ANALYSIS SRS (SOFTWARE REQUIREMENTS SPECIFICATION) USING SPECTRAL CLUSTERING

Name : Patricia Gertudis Manek
Student Identity Number : 5116201043
Supervisor : Daniel Oranova Siahaan, S.Kom, M.Sc,PD.Eng.

ABSTRACT

Software Requirements Specification (SRS) is a specification of functional (capabilities) and non-functional (constraints, characteristics, qualities, and properties) of a software that should be developed based on the conditions or capabilities that must equip the software or required by the users. For developers, SRS document is used as a reference in every stage of development. In the process of making SRS documents, there could be some defects. One of those defects, which were called as The Seven Sins of Specifier, is noise. Noise in the SRS could occur during the software requirements specification process. It lateralized in the form of information that is not relevant to the list user requirements.

This study aims to detect noise within the SKPL document. The noise detection uses spectral clustering to separate and group noise aside from the rest of requirements. In this method, each requirement statement is represented as a vector of frequencies of unique words appeared within the requirement statement. A cluster that has the widest average distance between its individual to its centroid is considered to have noise.

The experimentation shows that the proposed method has a high sensitivity, i.e. 1.0, for both real and synthetic data. Nevertheless, the method has a low specisivity for real data, i.e. 0.19, compare to the synthetic data, i.e. 0.83.

Keywords: Noise, Noise Detection, Requirements Specification, Spectral Clustering

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Tuhan Yesus Kristus atas anugerahNya sehingga penulis dapat menyelesaikan Tesis yang berjudul **“Analisis dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak) Menggunakan Kalsterisasi Spectral “**.

Pengerjaan tesis ini merupakan suatu kesempatan yang sangat berharga karena melalui pengerjaan tesis ini penulis mendapat banyak pengalaman untuk meningkatkan kemampuan yang telah didapatkan penulis, selama menempuh perkuliahan di Teknik Informatika ITS. terselesaikannya buku tesis ini, tidak terlepas dari bantuan dan dukungan semua pihak. Oleh karena itu pada kesempatan ini dengan kerendahan hati penulis mengucapkan terima kasih kepada semua pihak untuk semua bantuan yang telah diberikan, antara lain kepada:

1. Tuhan Yang Maha Esa atas anugerahNYA sehingga penulis dapat menyelesaikan tesis ini dengan baik.
2. Kedua orangtua penulis, bapa Agus Menek dan mama Agatha Usboko serta segenap keluarga besar Manek (kaka nona, kak sila, kaka Andi, adik epy, Efro, Irene dan Aldrich) yang telah memberikan dukungan berupa doa yang luar biasa yang selalu terus dipanjatkan, kasih sayang yang tulus dan tiada terbatas, selalu setia dan sabar dalam menghadapi keluh kesah penulis saat mengerjakan tesis ini.
3. Bapak Daniel Oranova Siahaan, S.Kom, M.Sc, PD.Eng selaku dosen pembimbing yang telah memberikan kepercayaan, motivasi, bimbingan, nasehat, perhatian serta semua bantuan yang telah diberikan kepada penulis dalam menyelesaikan Tesis ini.
4. Bapak Prof. Drs.Ec.Ir. Riyanarto Sarno, M.Sc.,Ph.D, Dr.Eng. bapak Radityo Anggoro, S.Kom, M.Sc dan Ibu Dr. Ir. Siti Rochimah, M.T., selaku dosen penguji yang telah memberikan bimbingan, saran, arahan, dan koreksi dalam pengerjaan tesis ini.

5. Bapak Waskitho Wibisono, S.Kom., M.Eng., PhD selaku ketua program pascasarjana Teknik Informatika ITS, dan Dr. Eng. Chastine Fatichah, S.Kom., M.Kom terimakasih untuk dukungannya.
6. Bapak Agus Zainal Arifin, S.Kom., M.Kom., selaku dosen wali yang selalu menyemangati dalam setiap konsultasi.
7. Mbak Lina, dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
8. Teman-teman penulis yang selalu memberikan semangat, dukungan dan hiburan kepada penulis, (Herna, Alifia, Adhy, Ulum, Bu Mirna, Kelly Ilmi, Amel, Eva) serta semua teman-teman angkatan Pasca Sarjana Teknik Informatika 2016
9. Teman-teman crue orang enak, special buat Herna trimakasih sudah jadi bagian dalam hidup ku juga, terus bertumbuh dan jadi berkat buat semua orang, juga buat dedy, even, kristo, budi Tuhan memberkati.
10. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu disini yang telah membantu terselesaikannya Tesis ini. Tuhan memberkati.

Sebagai manusia biasa, penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan dan memiliki banyak kekurangan. Sehingga dengan segala kerendahan hati, penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, 25 Oktober 2018

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN	v
ABSTRAK.....	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan.....	2
1.4. Manfaat.....	3
1.5. Kontribusi Penelitian.....	3
1.6. Batasan Masalah	3
BAB 2 KAJIAN PUSTAKA	5
2.1 Kebutuhan Perangkat Lunak	5
2.2 Kebutuhan Pengguna	7
2.3 Dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak)	7
2.4 <i>Noise</i> atau Derau Pada Dokumen SKPL (Spesifikasi Perangkat Lunak)	9
2.5 Pengolahan Bahasa Alamiah	10
2.6 Klasterisasi	11
2.7 Kakas Bantu Pemrosesan Bahasa Alamiah	14
BAB 3 METODE PENELITIAN	17

3.1. Studi Literatur	17
3.2. Desain dan Implementasi.....	18
3.2.1 Kebutuhan Perangkat Lunak di Dokumen SKPL	18
3.2.2 Pre-Processing Text.....	19
3.3 Pengujian.....	28
BAB 4 HASIL DAN ANALISIS	31
4.1 Pengumpulan Dataset.....	31
4.2 Proses Dataset Awal	32
4.3 Skenario Pengujian	33
4.4 Hasil Uji Skenario.....	35
4.4.1 Hasil Uji Skenario 1	35
4.4.2 Hasil Uji Skenario 2.....	41
4.5. Analisis Hasil Skenario 1 dan Skenario 2	44
BAB 5 PENUTUP	49
5.1 Kesimpulan.....	49
5.2 Saran.....	49
DAFTAR PUSTAKA	51

DAFTAR GAMBAR

Gambar 2.1. Daur Hidup Perangkat Lunak.....	5
Gambar 2.2 Tahapan Rekayasa Kebutuhan	6
Gambar 2.3 Diagram Alir Algoritma klasterisasi <i>k-means</i>	12
Gambar 2.4 Diagram Alir Algoritma Klasterisasi <i>Spectral</i>	14
Gambar 3.1 Alur Metodologi Penelitian.....	17
Gambar 3.2. Alur Proses Metode	18
Gambar 3.3 Alur Diagram Pra-pemrosesan Teks.....	19
Gambar 3.4 Visualisasi Hasil Identifikasi <i>Noise</i>	27
Gambar 4.1 Matriks Affinity dari dataset DA-1.	44

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Contoh Kebutuhan Sistem Serta Kebutuhan Penggunaanya	7
Tabel 2.2 Komparasi Penelitian <i>Noise</i> Sebelumnya.....	9
Tabel 2.3. Contoh Kebutuhan Perangkat Lunak	10
Tabel 3.1 Matriks S.....	22
Tabel 3.2 Contoh salah satu nilai kemiripan untuk kebutuhan R1-R2	23
Tabel 3.3 Matriks A (<i>Affinity</i>)	24
Tabel 3.4 Matriks D (<i>diagonal</i>)	24
Tabel 3.5 Matriks L (<i>laplacian</i>)	25
Tabel 3.6 Matriks Normalisasi	25
Tabel 3.7 Matriks U	26
Tabel 3.8 Matriks T	26
Tabel 3.9. Menentukan nilai k	27
Tabel 4.1 Dataset Penelitian.....	31
Tabel 4.2 Contoh Dataset Asli	32
Tabel 4.3 Daftar Dataset Sintetis.....	33
Tabel 4.4 Contoh Dataset Sintetis	34
Tabel 4.5 Hasil Uji Skenario 1	36
Tabel 4.6 Contoh Derau Hasil Uji Skenario 1 (Rata-rata hasil jarak terbesar)	37
Tabel 4.7 Contoh Deteksi Derau Hasil Uji Skenario 1 (Rata-rata terkecil).....	39
Tabel 4.8 Hasil Uji Coba Skenario 2	41
Tabel 4.9 Contoh Hasil Deteksi Derau Skenario 2 (rata-rata terbesar)	43
Tabel 4.10 Contoh Hasil Deteksi Derau Skenario 2 (rata-rata terkecil).....	43
Tabel 4.11 Nilai Reliabilitas untuk Dataset Asli	45
Tabel 4.10 Kinerja Rata-Rata Metode Deteksi Derau	46
Tabel 4.13 Kinerja Rata-Rata Metode Deteksi Derau	47
Tabel 4.14 Kinerja Rata-Rata Metode Deteksi Derau	47

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai beberapa hal dasar dalam pembuatan tesis penelitian yang meliputi latar belakang, perumusan masalah, tujuan, manfaat, kontribusi penelitian, dan batasan masalah.

1.1. Latar Belakang

Rekayasa kebutuhan (software engineering) adalah suatu proses mewujudkan serangkaian layanan yang dibutuhkan oleh pengguna atas suatu sistem dan batasan-batasan yang harus dipenuhi ketika dibangun maupun dioperasikan (Sommerville, 2007). Penelusuran, pengumpulan dan penyajian segala kebutuhan pengguna kedalam sebuah Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) adalah tahap awal dalam pengembangan perangkat lunak. Karena dokumen Spesifikasi Perangkat Lunak (SKPL) memiliki peran yang sangat penting dalam menghasilkan perangkat lunak yang berkualitas tinggi.

Penjaminan Kualitas Perangkat Lunak (PKPL) merupakan sebuah proses yang memantau seluruh proses pengembangan perangkat lunak untuk memastikan bahwa proses-proses tersebut dapat menghasilkan perangkat lunak yang berkualitas. Proses Penjamin Kualitas Perangkat Lunak (PKPL) mencakup seluruh tahap pada Daur Hidup Perangkat Lunak (DHPL). Proses PKPL, khususnya pada tahap spesifikasi kebutuhan, dinilai sangat penting menurut Boehm dan Basili (Meyer, 1985), untuk mencegah peningkatan biaya yang diperlukan saat pembangunan perangkat lunak. Perangkat lunak yang telah selesai diperbaiki memerlukan biaya lebih besar daripada pada saat tahap spesifikasi kebutuhan dan desain perangkat lunak.

Dalam melakukan spesifikasi kebutuhan terdapat tiga metode pendekatan yang dapat digunakan yaitu natural language, semi-formal language, dan formal language. Natural language adalah pendekatan spesifikasi kebutuhan yang menggunakan bahasa sehari-hari, semi-formal language menggunakan bahasa grafikal yang disertai penjelasan teks, dan formal language menggunakan konsep matematika seperti finite-state machines. Menurut survei yang pernah dilakukan Rossi pada tahun 1999 [2], sebanyak 71,8% dokumen

spesifikasi kebutuhan perangkat lunak dibuat dengan menggunakan pendekatan natural language. Hal ini disebabkan karena pendekatan tersebut lebih mudah dipahami oleh banyak pihak.

Meyer menjelaskan bahwa proses spesifikasi kebutuhan perangkat lunak yang menggunakan bahasa natural memiliki kekurangan jika dibandingkan dengan bahasa formal (Meyer, 1985). Hal tersebut disebabkan karena spesifikasi kebutuhan yang menggunakan natural language memiliki tujuh kesalahan umum yang sering dilakukan oleh pengembang perangkat lunak yang dirumuskan oleh Meyer sebagai *The seven sins of specifier*. Salah satu dari *The seven sins of specifier* adalah *noise* atau derau. Derau dapat muncul pada proses spesifikasi kebutuhan di dalam dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). *Noise* atau derau terjadi jika dalam spesifikasi kebutuhan, pengembang perangkat lunak menyertakan suatu informasi yang tidak relevan dengan keseluruhan kebutuhan perangkat lunak. Hal tersebut dapat membingungkan pembaca dan dapat berakibat buruk pada tahap-tahap pengembangan perangkat lunak selanjutnya.

1.2. Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut.

1. Bagaimana melakukan klasterisasi pernyataan kebutuhan dari suatu dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak)?
2. Bagaimana karakteristik dari hasil klasterisasi pernyataan kebutuhan dari suatu dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak)?
3. Apakah *noise* atau derau dapat dideteksi berdasarkan hasil klasterisasi pernyataan kebutuhan tersebut?

1.3. Tujuan

Tujuan yang akan dicapai dalam pembuatan tesis ini adalah menganalisis dokumen SKPL dengan menggunakan proses klasterisasi pernyataan kebutuhan menggunakan metode klasterisasi *spectral*.

1.4. Manfaat

Manfaat dari penelitian ini adalah memberikan informasi bagi pengembang terkait karakteristik dari kumpulan pernyataan kebutuhan dalam suatu dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak) Informasi yang mungkin muncul terkait ada atau tidaknya pernyataan kebutuhan yang tergolong *noise* atau derau. Diharapkan nantinya kualitas dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) akan lebih baik.

1.5. Kontribusi Penelitian

Kontribusi penelitian ini adalah penggunaan metode klasterisasi *spectral* dalam menganalisis dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL).

1.6. Batasan Masalah

1. Data dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) yang digunakan berbahasa inggris.
2. Pernyataan kebutuhan dinyatakan dalam bahasa alamiah.

(Halaman ini sengaja dikosongkan)

BAB 2

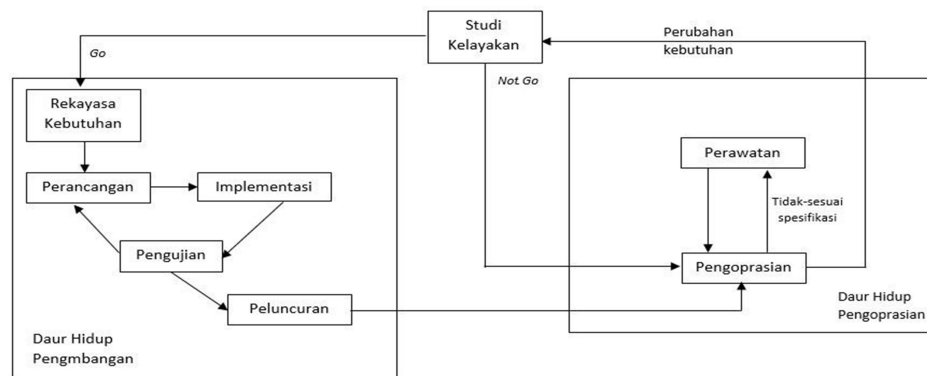
KAJIAN PUSTAKA

Pada bab ini akan dijelaskan tentang pustaka yang terkait dengan landasan penelitian. Pustaka yang terkait dianalisa dan dirangkum dalam bentuk studi komparasi.

2.1 Kebutuhan Perangkat Lunak

Kebutuhan dalam perangkat lunak dapat diartikan sebagai sekumpulan deskripsi tentang bagaimana suatu sistem berperilaku sehingga dapat memiliki nilai guna bagi penggunaanya (Siahaan, 2012). Kebutuhan tersebut dispesifikasikan dalam tahap rekayasa kebutuhan.

Daur Hidup Perangkat Lunak dapat dilihat pada Gambar 2.1 yang menunjukkan bahwa dalam kebutuhan perangkat lunak terdapat dua siklus kehidupan utama yaitu daur hidup pengembangan dan daur hidup pengoperasian perangkat lunak, yang dihubungkan oleh studi kelayakan dan proses peluncuran.

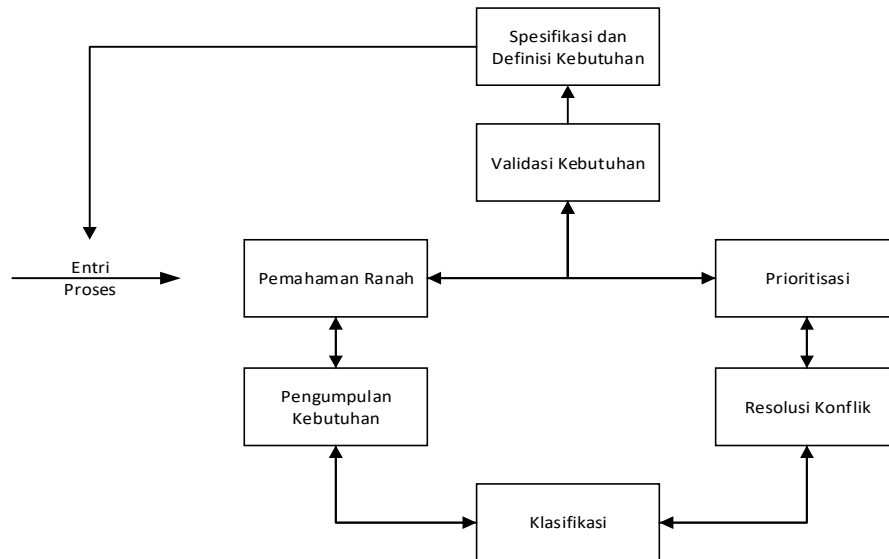


Gambar 2.1. Daur Hidup Perangkat Lunak

Terdapat dua siklus kehidupan dari suatu perangkat lunak, yaitu daur hidup pengembangan dan daur hidup pengoperasian. Dalam membuat sebuah aplikasi kita harus mengetahui kebutuhan dari aplikasi tersebut. Setelah mengetahui kebutuhannya, kita dapat merancang aplikasi tersebut. Dari perancangan kita dapat mengimplementasikan. Hasil dari implementasi selanjutnya diuji sampai benar-benar bisa digunakan sebelum pada akhirnya

diluncurkan. Setelah daur hidup pengembangan dilakukan sampai peluncuran, kita masuk ke daur hidup pengoperasian yaitu pengoperasian dan perawatan (Siahaan, 2012).

Proses rekayasa kebutuhan dilakukan dalam beberapa tahap seperti yang ditunjukkan pada Gambar 2.2 Proses dimulai dengan mempelajari tentang ranah lingkup permasalahan pada ranah pemahaman. Dari pemahaman ranah tersebut dikumpulkan kebutuhan perangkat lunak dan masuk pada klasifikasi dan diakhiri dengan pemeriksaan kembali kebutuhan-kebutuhan yang telah didapatkan terhadap relevansi dengan permasalahan utama. Kebutuhan-kebutuhan perangkat lunak yang telah berhasil dikumpulkan dirumuskan dalam sebuah dokumen SKPL (Siahaan, 2012).



Gambar 2.2 Tahapan Rekayasa Kebutuhan

2.2 Kebutuhan Pengguna

Kebutuhan pengguna adalah pernyataan atau gambaran pelayanan yang disediakan oleh sistem dan batasan-batasan bagi sistem yang akan dibangun agar pengguna dapat memenuhi tujuan penggunaan sistem.

Secara umum, kebutuhan perangkat lunak dapat dibagi ke dalam dua jenis yaitu : fungsional dan non-fungsional. Kebutuhan fungsional adalah kebutuhan yang dilihat dari sudut pandang pengguna. Kebutuhan non-fungsional adalah kebutuhan yang dilihat dari sudut pandang sistem. Selain dapat dikelompokkan ke dalam kebutuhan fungsional dan non-fungsional, kebutuhan juga dapat dikelompokkan lebih rinci lagi berdasarkan tingkat dari stakeholder (Siahaan, 2012).

Tabel 2.1 Contoh Kebutuhan Sistem Serta Kebutuhan Penggunaanya

Sistem	Kebutuhan Pengguna
SIM Akademik	<ul style="list-style-type: none">- Mahasiswa hendak menyusun rencana kuliah semester ini- Orangtua hendak melihat prestasi belajar anak didik

2.3 Dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak)

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) merupakan keluaran dari tahap rekayasa kebutuhan di dalam Daur Hidup Perangkat Lunak seperti pada Gambar 2.1. Dokumen tersebut berisi tentang kumpulan kebutuhan-kebutuhan perangkat lunak yang akan dibuat. Dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak) berfungsi sebagai landasan proses pengembangan perangkat lunak yang telah disetujui oleh pihak pengembang dan klien.

Terdapat empat pendekatan dalam menyusun dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak) yaitu :

1. Bahasa Alamiah

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) disusun dengan menggunakan bahasa sehari-hari.

2. Bahasa Alamiah Terstruktur

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) disusun dengan menggunakan bahasa sehari – hari yang ditata sesuai dengan struktur tertentu.

3. Bahasa Semi-formal

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) disusun dengan menggunakan bahasa grafikal yang dilengkapi dengan penjelasan teks.

4. Bahasa Formal

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) disusun sepenuhnya dengan notasi-notasi matematika seperti *finite state machine*.

Menurut survei yang telah dilakukan sebelumnya, pendekatan bahasa alamiah paling banyak digunakan untuk menyusun dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) sebesar 71,8% (Rossi, 1999). Hal tersebut disebabkan karena pendekatan bahasa alamiah lebih mudah dipahami dan diimplementasi. Namun karena sifatnya yang subjektif, penggunaan bahasa alamiah dapat menimbulkan kerancuan dan kesalahan-kesalahan lainnya yang telah dibahas oleh Meyer dalam “The seven sins of specifier” (Meyer, 1985).

Meyer mendeskripsikan secara mendalam tujuh kesalahan umum yang sering dilakukan pengembang perangkat lunak sebagai “The seven sins of specifier” salah satunya adalah noise, noise dapat muncul pada proses spesifikasi kebutuhan yaitu menyertakan suatu informasi yang tidak relevan dengan keseluruhan kebutuhan perangkat lunak.

2.4 *Noise* atau Derau Pada Dokumen SKPL (Spesifikasi Perangkat Lunak)

Spesifikasi dalam rekayasa perangkat lunak adalah suatu aktivitas untuk mengumpulkan kebutuhan dalam pembuatan perangkat lunak. Kumpulan kebutuhan tersebut dirumuskan dalam dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak). Namun terkadang terdapat suatu kebutuhan yang tidak relevan dengan kebutuhan lainnya. Kebutuhan tersebut dapat mengurangi kualitas informasi yang diberikan oleh dokumen Kebutuhan Perangkat Lunak (SKPL) dan dapat dianggap sebagai *noise*, seperti yang telah dijelaskan oleh Meyer (Meyer, 1985). Terdapat beberapa penelitian terkait yang telah dikembangkan tentang deteksi *noise* atau derau, seperti yang terdapat pada tabel 2.2.

Tabel 2.2 Komparasi Penelitian *Noise* Sebelumnya

No	Judul	Masalah	Metode	Hasil
1	Simultaneous Clustering and Noise Detection for Theme-based Summarization (Cai <i>et al.</i> , 2011)	Mendeteksi <i>noise</i> berupa kalimat dalam dokumen teks di internet	Menggunakan <i>spectral clustering</i>	Berupa <i>noise</i> pada teks
2.	Detecting Deviations in Text Collection : An Approach Using Conceptual Graphs (Montes-y-gómez, Gelbukh and López-lópez, 2002)	Mendeteksi penyimpangan teks (paper) “computer science”.	Menggunakan Graphs Conceptual	Berupa visualisasi graph terhadap penyimpangan yang terjadi pada suatu dokumen teks.

Pada Tabel 2.3 contoh data yang akan digunakan untuk menguji pendeteksi *noise* atau derau pada dokumen SKPL (Spesifikasi Kebutuhan Perangkat Lunak) yang akan dibuat. Tabel tersebut berisi tentang kebutuhan sebuah perangkat lunak. Kebutuhan R-01 sampai kebutuhan R-04 dianggap bukan *noise* atau derau karena masih membahas mengenai dokumen proyek pelajar sedangkan kebutuhan R-05 dianggap *noise* atau derau karena kebutuhan tersebut tidak relevan, kebutuhan tersebut membahas mengenai dokumen sistem pengarsipan. Penelitian yang akan dilakukan bertujuan untuk dapat mendeteksi kebutuhan tersebut secara otomatis.

Tabel 2.3. Contoh Kebutuhan Perangkat Lunak

ID	Requirement Statement	Noise
R-01	System Administrator can list all students in different period of different group to check any error	No
R-02	System Administrator can reset the student's password if required	No
R-03	Administration Staff can list all students in different period of different group	No
R-04	Administration Staff can make a school announcement	No
R-05	The archive administrator cannot access documents.	Yes

2.5 Pengolahan Bahasa Alamiah

Salah satu teknik pemrosesan bahasa alami yang seringkali digunakan pada pra proses sebagai berikut:

1. *Tokenisasi*

Pada proses tokenisasi, teks dokumen dipecah menjadi bagian-bagian berupa kalimat, paragraf, atau dokumen.

2. *Lowercase*

Merubah semua huruf menjadi huruf kecil.

3. *Stopword Removal*

Proses untuk menghilangkan angka, simbol, dan kata ganti

4. *Stemming*

Menghilangkan imbuhan dari kata-kata yang berimbuhan sehingga, hanya ada kata-kata yang penting yang tersisa dalam teks.

2.6 Klasterisasi

Klasterisasi adalah pengelompokan data ke dalam sejumlah kelompok, berdasarkan kesamaan karakteristik masing-masing data pada kelompok-kelompok yang ada. Sehingga kelompok dengan karakteristik yang sama (memiliki hubungan) akan berada pada satu kelompok yang sama. Sedangkan kelompok yang memiliki karakteristik berbeda (tidak ada hubungan) akan dikelompokkan dalam kelompok lain (klaster lain). Klasterisasi juga dikelompokkan sebagai metode pembelajaran tidak terbimbing (*unsupervised learning*) (Prasetyo, 2014).

Klasterisasi bertujuan untuk mengelompokkan data dengan karakteristik yang sama ke suatu kelompok yang sama dan data dengan karakteristik yang berbeda ke kelompok yang lain. Pada umumnya terdapat dua pendekatan klasterisasi yaitu pendekatan partisi dan pendekatan hirarki.

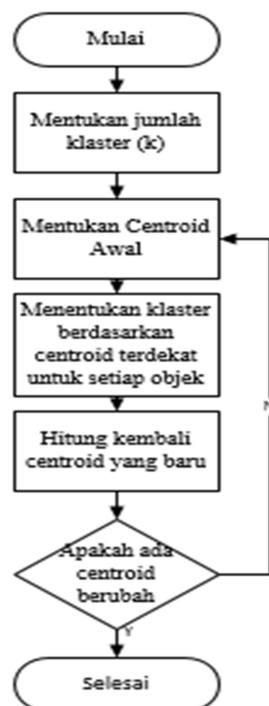
Klasterisasi hirarkikal dokumen teks merupakan metode pengelompokan dokumen yang bekerja dengan membangun sebuah hirarki kelompok dokumen atau klaster. Hirarki klaster dapat disebut juga sebagai dendrogram. Klasterisasi hirarkikal telah digunakan untuk pembentukan taksonomi konsep pada suatu teks (Knijff, Frasincar and Hogenboom, 2013). Kelebihan klasterisasi hirarkikal dibandingkan dengan metode klasterisasi lain adalah tingkat hirarki yang dapat ditentukan sesuai kebutuhan (Popat and Emmanuel, 2014). Contoh metode klaster dengan pendekatan hirarki adalah klasterisasi hirarkikal *agglomerative*.

Klasterisasi dengan pendekatan partisi merupakan pengelompokan data dari satu kelompok besar kemudian dibagi menjadi beberapa kelompok yang lebih kecil. Contoh metode klasterisasi dengan pendekatan partisi adalah klasterisasi k-means dan klasterisasi *spectral* (Prasetyo, 2014).

K-Means merupakan metode klusterisasi yang paling umum digunakan untuk mengelompokkan dokumen (Saiyad, Prajapati and Dabhi, 2016). Metode klusterisasi ini memiliki kekurangan dimana jumlah klaster perlu diketahui terlebih dahulu (Gupta and Srivastava, 2014). Pada umumnya k-Means menggunakan pendekatan VSM (*Vector Space Model*), dimana dokumen dimodelkan dalam vektor yang memiliki kata sebagai fitur.

Masukkan atau input dari k-means adalah kumpulan dokumen **D** dan parameter jumlah klaster **k**. Tahapan algoritma k-means sebagai berikut :

1. Tentukan nilai k sebagai jumlah klaster
2. Pilih k data dari data set sebagai centroid
3. Alokasikan semua data ke centroid terdekat dan hitung matriks jarak
4. Hitung kembali centroid berdasarkan data yang mengikuti klaster masing-masing
5. Ulangi langkah 3 dan 4 sampai tidak ada dokumen yang berpindah klaster.

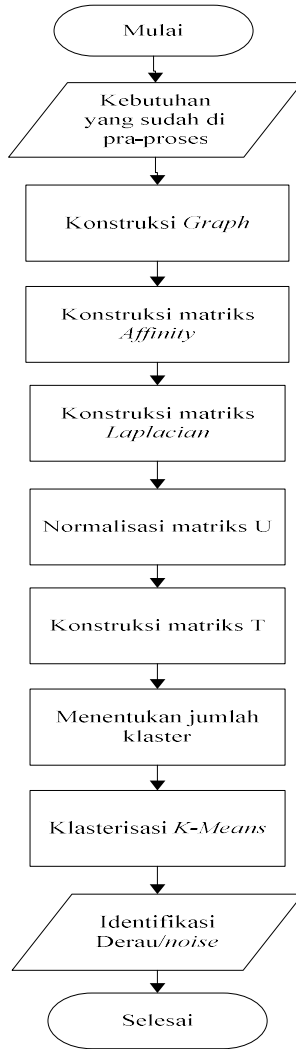


Gambar 2.3 Diagram Alir Algoritma klusterisasi k-means

Pada metode klasterisasi *spectral*, pengelompokan data didasarkan atas kesamaan antara setiap data. Pada klasterisasi *spectral* akan dibentuk sebuah *graph* dari data yang ada. Dimana *verteks* dari *graph* tersebut merupakan setiap *record* pada data. *Edge*-nya berupa hubungan antar data yang biasanya bernilai jarak dari dua *record* (Uw *et al.*, 2001).

Langkah-langkah algoritma klasterisasi *spectral* sebagai berikut :

1. Dari data set dibentuk *graph* tidak berarah dalam bentuk matriks. dimana $G=(S,A)$
2. Dari *graph* tersebut dibentuklah matriks *affinity* dimana matriks tersebut merupakan nilai bobot dari semua *edge* yang menghubungkan satu *node* dengan *node* yang lain.
3. Dari matriks *affinity* dihitung matriks *laplacian*
4. Menghitung normalisasi matriks *laplacian*
5. Kemudian dihitung *eigen vector* dan *eigen value* sehingga membentuk matriks U
6. Normalisasi matriks U sehingga membentuk k kolom yang merepresentasikan setiap normalisasi nilai matriks U pada setiap kolomnya
7. Menentukan nilai k
8. Hasil dari data normalisasi matriks U, kemudian diklaster dengan klasterisasi *k-means*
9. Inisialisasi, tentukan nilai k sebagai jumlah klaster
10. Pilih k data dari data set sebagai *centroid*
11. Alokasikan semua data ke *centroid* terdekat dan hitung matriks jarak
12. Hitung kembali *centroid* berdasarkan data yang mengikuti klaster masing-masing
13. Ulangi langkah 11 dan 12 sampai tidak ada dokumen yang berpindah klaster.
14. Setelah di klaster dengan klasterisasi *k-means* dimana klaster dengan jarak rata-rata anggota ke *centroid* yang paling kecil itu yang dianggap derau atau *noise*.



Gambar 2.4 Diagram Alir Algoritma Klasterisasi *Spectral*

2.7 Kakas Bantu Pemrosesan Bahasa Alamiah

1. Cosine Similarity

Cosine *similarity* merupakan salah satu metode pengukuran kemiripan antara dua dokumen atau teks. Pada *cosine similarity* dokumen atau teks dianggap sebagai vektor. Rumus *cosine similarity* (Garcia and Co, 2015):

$$\text{similarity} = \cos(\theta) = \frac{R_i \times R_j}{\|R_i\| \times \|R_j\|} = \frac{\sum_{k=1}^n (R_i \times R_j)}{\sqrt{\sum_{k=1}^n R_i^2} \times \sqrt{\sum_{k=1}^n R_j^2}} \quad (1.1)$$

Di mana R_i dan R_j adalah vektor *term frequency* dari dokumen, dengan nilai *cosine similarity* berada pada *range* 0-1.

2. Koefisien Kappa

Koefisien Kappa berfungsi untuk menghitung tingkat persetujuan antara dua penilai dalam memberikan penilaian pada suatu kasus (Cohen, 1960). Koefisien Kappa memperhitungkan probabilitas terjadinya persetujuan dan penilaian di antara kedua penilai. Nilai koefisien Kappa selalu diantara 0 dan 1, dengan nilai 1 menunjukkan kesempurnaan tingkat persetujuan.

Tabel 2.4. Masukan Perhitungan Nilai Koefisien Kappa

		Penilai 2			
		Nilai	A	B	Total
Penilai	1	A	n_aa	n_ab	n_a*
		B	n_ba	n_bb	n_b*
Total			n_*a	n_*b	

Pada Tabel 2.4 dapat dilihat bahwa terdapat 2 penilai yaitu Penilai 1 dan Penilai 2. Setiap penilai dapat memberikan dua kategori penilaian yaitu nilai a atau nilai b pada suatu kasus. Elemen dari Tabel 2.4 merupakan jumlah kejadian Penilai 1 dan Penilai 2 memberikan penilaian suatu kasus. Penjelasan variabel pada elemen Tabel 2.4 adalah sebagai berikut :

- n_{aa} adalah jumlah kejadian Penilai 1 dan Penilai 2 memberikan nilai a pada suatu kasus.
- n_{ab} adalah jumlah kejadian Penilai 1 memberikan nilai a dan Penilai 2 memberikan nilai b pada suatu kasus.
- n_{ba} adalah jumlah kejadian Penilai 1 memberikan nilai b dan Penilai 2 memberikan nilai a pada suatu kasus.
- n_{bb} adalah jumlah kejadian Penilai 1 dan Penilai 2 memberikan nilai b pada suatu kasus.

Nilai koefisien Kappa k dapat dihitung jika jumlah kejadian penilaian Penilai 1 dan Penilai 2 telah diketahui. Pertama – tama probabilitas persetujuan yang telah diobservasi antara kedua penilai, atau P_o dihitung dengan Persamaan (2.1).

$$P_o = \frac{n_{aa} + n_{bb}}{n_{aa} + n_{ab} + n_{ba} + n_{bb}} \quad (2.1)$$

Lalu probabilitas kedua penilai memberikan nilai a secara acak, atau P_a , dihitung, begitu juga dengan nilai P_b dengan Persamaan (2.2) dan (2.3). Sehingga nilai probabilitas persetujuan secara acak, atau P_e dapat dihitung dengan Persamaan (2.4).

$$P_a = \frac{n_{a*}}{n_{aa} + n_{ab} + n_{ba} + n_{bb}} \cdot \frac{n_{*a}}{n_{aa} + n_{ab} + n_{ba} + n_{bb}} \quad (2.2)$$

$$P_b = \frac{n_{b*}}{n_{aa} + n_{ab} + n_{ba} + n_{bb}} \cdot \frac{n_{*b}}{n_{aa} + n_{ab} + n_{ba} + n_{bb}} \quad (2.3)$$

$$P_e = P_a + P_b \quad (2.4)$$

Nilai koefisien $Kappa$ k dapat dihitung dengan Persamaan

$$k = \frac{P_o - P_e}{1 - P_e} \quad (2.5)$$

Hasil perhitungan nilai $Kappa$ tersebut kemudian dapat diinterpretasikan secara kuantitatif dengan menggunakan tabel interpretasi seperti pada Tabel 2.5

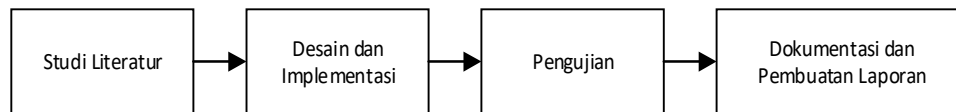
Tabel 2.5 Interpretasi Nilai Gwet AC1

Index Kappa	Proporsi Kesepakatan
< 0	Rendah
0.01 - 0.20	Sedikit
0.21 - 0.40	Cukup
0.41 - 0.60	Sedang
0.61 - 0.80	Substansial
0.81 - 1	Hampir sempurna

BAB 3

METODE PENELITIAN

Bab ini akan memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini, yang terdiri dari (1) studi literatur, (2) desain dan implementasi, (3) pengujian dan (4) dokumentasi dan pembuatan laporan. Ilustrasi alur metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Metodologi Penelitian

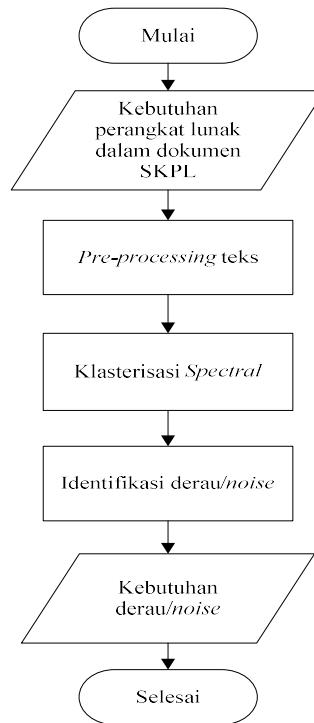
3.1. Studi Literatur

Tahap studi literatur bertujuan untuk mengumpulkan referensi-referensi yang dapat menunjang penelitian. Sumber referensi dapat berupa jurnal ilmiah atau buku teks. *Noise* atau derau pada dokumen SKPL dapat dianggap juga sebagai *outlier* atau pencilan pada sebuah dokumen teks. Sehingga referensi yang dikumpulkan berhubungan dengan *noise* di dalam dokumen SKPL dan metode pendeteksi *outlier* pada dokumen teks. Referensi tersebut digunakan untuk merumuskan permasalahan yang menjadi landasan dilakukannya penelitian ini dan solusi yang akan diusulkan. Berdasarkan studi literatur yang telah dilakukan, informasi yang berkaitan dengan penelitian yang dilakukan ini, seperti berikut :

1. *Noise* atau derau merupakan salah satu kesalahan yang sering dilakukan oleh analis kebutuhan dalam menyusun dokumen SKPL.
2. *Noise* adalah suatu elemen pada dokumen SKPL yang tidak relevan dengan informasi lainnya.
3. *Noise* atau derau dapat dianggap sebagai *outlier* atau pencilan pada dokumen teks.
4. Metode klasterisasi dapat digunakan sebagai pendeteksi *noise* pada dokumen SKPL.

3.2. Desain dan Implementasi

Tahap ini bertujuan untuk merancang dan mengimplementasi solusi yang diusulkan untuk mengatasi permasalahan yang telah dirumuskan sebelumnya. Metode yang akan diusulkan memiliki alur proses seperti pada Gambar 3.2.



Gambar 3.2. Alur Proses Metode

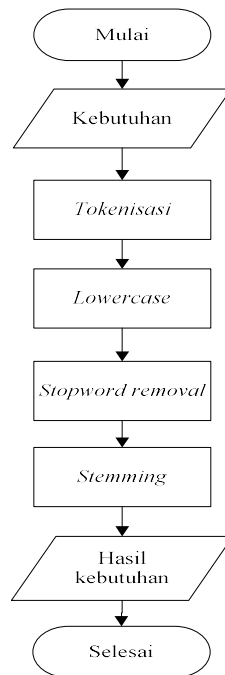
3.2.1 Kebutuhan Perangkat Lunak di Dokumen SKPL

Masukan dari metode deteksi *noise* yang akan dibuat adalah kebutuhan-kebutuhan perangkat lunak yang tertera pada dokumen SKPL. Kebutuhan tersebut diekstraksi secara manual dari dokumen SKPL. Setiap dokumen SKPL akan menghasilkan sebuah set kebutuhan perangkat lunak.

Setiap kebutuhan yang telah diekstraksi akan diberikan label berupa nilai *Boolean* yaitu 1 atau 0. Kebutuhan akan berlabel 1 jika kebutuhan tersebut dianggap sebagai *noise*, sisanya akan berlabel 0. Pemberian label akan dilakukan secara manual oleh penulis. Contoh kebutuhan yang telah diekstraksi dari dokumen SKPL dan labelnya dapat dilihat pada Tabel 2.2.

3.2.2 Pre-Processing Text

Pre-Processing Text atau Pra-pemrosesan teks akan diimplementasi pada setiap kebutuhan perangkat lunak.



Gambar 3.3 Alur Diagram Pra-pemrosesan Teks

Pada tahap pra-pemrosesan teks terdapat tiga proses utama, berikut adalah Contoh hasil pra-pemrosesan teks pada kebutuhan perangkat lunak :

1. *Tokenisasi*

Tokenisasi mempunyai peranan untuk memecah teks menjadi bagian unit terkecil dalam pemrosesan bahasa atau disebut dengan potongan kata. Dokumen teks

R1 : Submit jobs with the associated deadline cost and execution time

R2 : Query the cluster to establish the current cost per unit time for submitting new job

R3 : Monitor the status of submitted jobs

R4 : Cancel job submitted by him

R5 : Requirements engineer can remove a document

Akan dipecah menjadi :

R1 : Submit// jobs// with// the// associated// deadline// ,// cost// and// execution
//time//

R2 : Query// the// cluster// to// establish// the// current// cost// per// unit// time//
for// submitting// new// job//

R3 : Monitor// the// status// of// submitted// jobs//

R4 : Cancel// job// submitted// by// him//

R5 : Requirements// engineer// can// remove// a// document//

2. *Lowercase* (Konversi menjadi huruf kecil)

Merubah semua huruf pada teks dokumen menjadi huruf kecil. Pada teks dokumen sebelumnya

R1 : Submit// jobs// with// the// associated// deadline// ,// cost// and// execution
//time//

R2 : Query// the// cluster// to// establish// the// current// cost// per// unit// time//
for// submitting// new// job//

R3 : Monitor// the// status// of// submitted// jobs//

R4 : Cancel// job// submitted// by// him//

R5 : Requirements// engineer// can// remove// a// document//

Akan berubah menjadi

R1 : submit// jobs// with// the// associated// deadline// ,// cost// and// execution
//time//

R2 : query// the// cluster// to// establish// the// current// cost// per// unit// time//
for// submitting// new// job//

R3 : monitor// the// status// of// submitted// jobs//

R4 : cancel// job// submitted// by// him//

R5 : requirements// engineer// can// remove// a// document//

3. *Stopword Removal* (Eliminasi angka, simbol, dan kata ganti)

Proses ini berfungsi untuk menghilangkan angka simbol dan kata ganti.

Dokumen sebelumnya

R1 : submit// jobs// with// the// associated// deadline// ,// cost// and// execution
//time//

R2 : query// the// cluster// to// establish// the// current// cost// per// unit// time//
for// submitting// new// job//

R3 : monitor// the// status// of// submitted// jobs//

R4 : cancel// job// submitted// by// him//

R5 : requirements// engineer// can// remove// a// document//

akan berubah menjadi

R1 : submit// jobs// associated// deadline// cost// execution //time//

R2 : query// cluster// establish// current// cost// per// unit// time// submitting//
new// job//

R3 : monitor// status// submitted// jobs//

R4 : cancel// job// submitted//

R5 : requirements// engineer// remove// document//

4. *Stemming*

Stemming memiliki peran untuk menghilangkan kata-kata yang berimbuhan menjadi kata dasar. Dokumen sebelumnya

R1 : submit// jobs// associated// deadline// cost// execution //time//

R2 : query// cluster// establish// current// cost// per// unit// time// submitting//
new// job//

R3 : monitor// status// submitted// jobs//

R4 : cancel// job// submitted//

R5 : requirements// engineer// remove// document//

akan berubah menjadi

R1 : submit// job// associate// deadline// cost// execution //time//
R2 : query// cluster// establish// current// cost// per// unit// time// submit// new//
job//
R3 : monitor// status// submit// job//
R4 : cancel// job// submit//
R5 : requirement// engineer// remove// document//

3.2.3 Klasterisasi *Spectral*

Metode klasterisasi *spectral* seperti yang sudah dijelaskan sebelumnya, merupakan pengelompokkan data didasarkan atas kesamaan antara setiap data. Kesamaan tersebut dapat dilihat dari keterkaitan antara setiap data.

Langkah-langkah dalam melakukan Algoritma *Spectral Clustering* sebagai berikut :

1. Menyediakan kebutuhan yang telah di *pre-processing*.
Contoh kebutuhan :
R1 : submit job associate deadline cost execution time
R2 : query cluster establish current cost per unit time submit new job
R3 : monitor status submit job
R4 : cancel job submit
R5 : requirement engineer remove document
2. Membentuk *graph* tak berarah $G = \{S,A\}$. Di mana S merupakan himpunan *node*. *Node* adalah kalimat dalam bentuk fitur vektor dengan ukuran m , dimana m adalah jumlah kata unik. Sedangkan n adalah jumlah kalimat dalam data set dan A adalah nilai bobot *edge* yang menghubungkan kalimat yang satu dengan yang lain (Cai *et al.*, 2011).

Tabel 3.1 Matriks S

	Submit	job	associate	Deadline	Cost	Execution	time	query	cluster	Establish
R1	1	1	1	1	1	1	1	0	0	0
R2	1	1	0	0	1	0	1	1	1	1
R3	1	1	0	0	0	0	0	0	0	0
R4	1	1	0	0	0	0	0	0	0	0
R5	0	0	0	0	0	0	0	0	0	0

	current	per	unit	New	monitor	Status	Cancel	requirement	engineer	remove	document
R1	0	0	0	0	0	0	0	0	0	0	0
R2	1	1	1	1	0	0	0	0	0	0	0
R3	0	0	0	0	1	1	0	0	0	0	0
R4	0	0	0	0	0	0	1	0	0	0	0
R5	0	0	0	0	0	0	0	1	1	1	1

3. Konstruksi matriks *affinity* $A [a_{ij}]$ merupakan matriks bobot dari semua *edge* yang menghubungkan satu *node* dengan *node* yang lain pada *graph* G . Nilai bobot dari A_{ij} dihitung berdasarkan nilai kemiripan antara kalimat R_i dan R_j . Perhitungan kemiripan ini didasarkan pada rumus *cosine similarity*.

$$Similarity = \cos(\theta) = \frac{R_i \times R_j}{\|R_i\| \times \|R_j\|} = \frac{\sum_{k=1}^n (R_i \times R_j)}{\sqrt{\sum_{k=1}^n R_i^2} \times \sqrt{\sum_{k=1}^n R_j^2}} \quad (3.1)$$

Pada nilai kemiripan R3 dan seterusnya dilakukan perhitungan yang sama, hasil hitungan dari *cosine similarity* yang sudah dihitung pada tahap sebelumnya direpresentasikan dalam bentuk matriks seperti yang ditampilkan pada Tabel 3.2.

Tabel 3.2 Contoh salah satu nilai kemiripan untuk kebutuhan R1-R2

R1	R2	R1.R2	kuadrat(^R1)	kuadrat(^R2)
1	1	1	1	1
1	1	1	1	1
1	0	0	1	0
1	0	0	1	0
1	1	1	1	1
1	0	0	1	0
1	1	1	1	1
0	1	0	0	1
0	1	0	0	1
0	1	0	0	1
0	1	0	0	1
0	1	0	0	1
0	1	0	0	1
0	1	0	0	1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

R1	R2	R1.R2	kuadrat(^R1)	kuadrat(^R2)
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
		4	7	11
akar jml R1^2		2,65		
akar jml R2^2		3,32		
perkalian akar (jml R1^2+jml R2^2)		8,77		
Hasil		0,46		

Tabel 3.3 Matriks A (*Affinity*)

Aij	R1	R2	R3	R4	R5
R1		0,46	0,38	0,44	0,00
R2	0,46		0,30	0,35	0,00
R3	0,38	0,30		0,58	0,00
R4	0,44	0,35	0,58		0,00
R5	0,00	0,00	0,00	0,00	

4. Langkah berikutnya adalah membentuk matriks *laplacian*.

Langkah pertama dalam membangun matriks Laplacian adalah :

- Menghitung matriks *diagonal* (D). Matriks diagonal (D) dibangun dari matriks *affinity* (A) yang sudah dihitung sebelumnya.

Perhitungan matriks D (*diagonal*) dapat dihitung dengan rumus :

$$d_i = \sum_{j=1}^n W_{ij} \quad (3.2)$$

Tabel 3.4 Matriks D (*diagonal*)

D	R1	R2	R3	R4	R5
R1	1,28	0,00	0,00	0,00	0,00
R2	0,00	1,11	0,00	0,00	0,00
R3	0,00	0,00	1,26	0,00	0,00
R4	0,00	0,00	0,00	1,37	0,00
R5	0,00	0,00	0,00	0,00	0,00

Contoh hitungan

Nilai diagonal R1 dan R1 nilainya 1,28 diperoleh dari hasil penjumlahan nilai dari kebutuhan R1-R5 pada matriks *affinity*. Perhitungan dengan cara yang sama dilakukan pada kebutuhan yang lain.

- Menghitung nilai matriks *laplacian*. Nilai matriks *laplacian* diperoleh dari pengurangan nilai pada matriks D (*diagonal*) dan matriks A (*affinity*) dengan rumus: $L = D - A$ (3.3)

Tabel 3.5 Matriks L (*laplacian*)

L	R1	R2	R3	R4	R5
R1	1,28	0,82	0,90	0,84	1,28
R2	0,65	1,11	0,81	0,76	1,11
R3	0,88	0,96	1,26	0,68	1,26
R4	0,93	1,02	0,79	1,37	1,37
R5	0,00	0,00	0,00	0,00	0,00

Contoh hitungan

Nilai *laplacian* R1 dan R2 = 0,82 diperoleh dari nilai baris matriks D (*diagonal*) R1- nilai kolom matriks A (*affinity*) R2. Perhitungan dengan cara yang sama dilakukan untuk kebutuhan yang lain.

5. Membentuk normalisasi matriks *laplacian*. Nilai normalisasi matriks *laplacian* dibangun dari matriks bobot *diagonal* (D) yang sudah dihitung sebelumnya dengan rumus berikut (Luxburg, 2007) :

$$L_{sym} = \begin{cases} 1 & \text{Jika } i = j \text{ dan } d_i \neq 0 \text{ maka nilainya } 1 \\ \frac{-1}{\sqrt{d_i+d_j}} & \text{Jika } i \neq j \text{ dan } d_i \text{ bertetangga dengan } d_j \\ 0 & \text{Jika tidak keduanya atau sebalik.} \end{cases} \quad (3.4)$$

Tabel 3.6 Matriks Normalisasi

Lsym	R1	R2	R3	R4	R5
R1	1,00	-0,65	-0,63	-0,61	-0,88
R2	-0,65	1,00	-0,65	-0,64	-0,95
R3	-0,63	-0,65	1,00	-0,89	-0,89
R4	-0,61	-0,64	-0,89	1,00	-0,88
R5	-0,88	-0,95	-0,89	-0,88	0,00

Contoh hitungan:

Nilai normalisasi *laplacian* kebutuhan R1 dan R2 = -0,65 diperoleh dari -1/ nilai kuadrat dari matriks *diaogonal* R1+nilai kuadrat dari matiks *diagonal* R2. Perhitungan dengan cara yang sama dilakukan pada kebutuhan yang lain. Setelah mendapat nilai dari setiap kebutuhan dibentuklah ke dalam matriks normalisasi seperti pada Tabel 3.6.

6. Konstruksi matriks U. Matriks U dibangun dengan menghitung *eigen vector* dari hasil matriks normalisasi *laplacian*. Maka terbentuklah matriks $U = \{u_1, u_2, \dots, u_k\}$, seperti pada tabel 3.7.

Tabel 3.7 Matriks U

U	U1	U2	U3	U4	U5
1	0,38	0,08	-0,74	-0,55	0,03
2	0,40	0,00	-0,40	0,83	0,00
3	0,00	0,48	0,28	-0,06	-0,72
4	0,41	0,49	0,32	-0,04	0,70
5	0,60	-0,72	0,33	-0,13	0,00

7. Langkah berikut adalah normalisasi data dengan nilai matriks U sehingga membentuk k kolom yang merepresentasikan setiap matriks normalisasi u pada setiap kolom. Maka terbentuklah matriks t . Perhitungan nilai matriks t didasarkan pada rumus (Luxburg, 2007):

$$t_{ij} = \frac{u_{ij}}{\sum_k \sqrt{u_{ik}^2}} \quad (3.5)$$

Dari hitungan yang didasarkan pada rumus tersebut akan membentuk matriks t .

Tabel 3.8 Matriks T

T	1	2	3	4	5
y1	0,38	0,08	-0,74	-0,55	0,03
y2	0,40	0,00	-0,40	0,83	0,00
y3	0,00	0,53	0,31	-0,07	-0,79
y4	0,41	0,49	0,32	-0,04	0,70
y5	0,60	-0,72	0,33	-0,13	0,00

Contoh hitungan:

Nilai 0,38 diperoleh dari nilai y1/kuadrat dari nilai normalisasi u_1, u_2, u_3, u_4 dan u_5 , perhitungan dengan cara yang sama dilakukan

untuk kebutuhan yang lainnya. Seperti pada Tabel 3.8.

8. Menentukan nilai k (*Cluster*)

Dari perhitungan normalisasi matriks U yang telah dihitung sebelumnya, dapat menentukan nilai k yang kemudian akan diklaster dengan *k-mens clustering*. Perhitungan nilai k didasarkan pada rumus (Cai *et al.*, 2011) :

$$k = \operatorname{argmax} \{ \lambda_{k+1}(L) - \lambda_k(L) \} \quad (3.6)$$

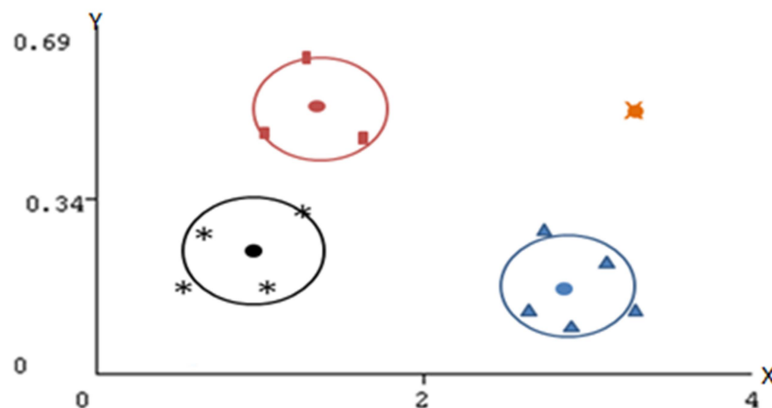
Tabel 3.9. Menentukan nilai k

y1	y2	y3	y4	y5
-2,39	1,31	1,54	1,66	1,89
-3,70	C1			
-0,23	C2			
-0,12	C3			
-0,23	C4			
berdasarkan rumus tersebut terdapat tiga cluster. nilai cluster yang besar sebagai nilai $k = 4,00$				

Contoh hitungan :

Nilai -3,70 diperoleh dari $(-2,39) - 1,31 = 4,00$

9. Hasil dari data normalisasi matriks U , kemudian di-*cluster* dengan *K-means clustering* dimana klaster dengan jarak rata-rata anggota ke *centroid* yang paling kecil itulah yang dianngap derau atau pencilan.



Gambar 3.4 Visualisasi Hasil Identifikasi *Noise*

3.2.4 Identifikasi Pencilan

Hasil dari klasterisasi *spectral* adalah beberapa klaster yang berisi kebutuhan perangkat lunak. Untuk menentukan klaster derau atau noise dapat dilakukan dengan 2 cara yaitu :

1. **Metode A** : penentuan klaster derau atau *noise* dengan memilih klaster yang memiliki rata-rata jarak ke *centroid* yang kecil.
2. **Metode B** : penentuan klaster derau atau *noise* dengan memilih klaster yang jarak rata-rata ke *centroid* yang besar.

Implementasi metode akan menggunakan bahasa pemrograman Python versi 2.7. Pustaka-pustaka pendukung yang akan digunakan dalam implementasi metode adalah pustaka numpy, scipy, scikit-learn, panda dan NLTK.

3.3 Pengujian

Data uji diperoleh dari dataset yang ada berupa kebutuhan-kebutuhan di dalam sebuah dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Sebelumnya, kebutuhan-kebutuhan tersebut diberi label berupa nilai boolean 1 atau 0. Label 1 menandakan bahwa kebutuhan tersebut dianggap noise oleh penulis. Pelabelan data uji tersebut merepresentasikan penilaian noise oleh penulis.

Pada pengujian, data uji seperti pada Tabel 2.3 digunakan sebagai masukan untuk pendeteksi derau. Pendeteksi noise akan menghasilkan prediksi kebutuhan yang dianggap sebagai derau. Kedua hasil derau dari pendeteksi derau dan penilaian penulis akan dibandingkan menggunakan kaskas bantu yaitu menghitung nilai indeks koefisien Kappa, seperti yang telah dijelaskan pada subbab sebelumnya. Nilai dari koefisien Kappa dapat merepresentasikan tingkat konsistensi hasil derau yang diberikan oleh pendeteksi derau dengan penilaian penulis.

Adapun skenario pengujian uji coba yang dilakukan sebagai berikut :

1. Mengekstraksi pernyataan kebutuhan dari setiap dokumen SKPL yang ada ke dalam format csv, seperti contoh: KODE; PERNYATAAN_KEBUTUHAN; LABEL. Setiap SKPL akan disimpan ke dalam satu buah berkas csv. Setiap berkas csv merupakan satu buah data set pengujian. Berikut contoh berkas csv dari sebuah SKPL.
R01; Operator dapat membuka berkas.; 0
R02; Mahasiswa dapat memilih mata kuliah yang hendak diambil.; 0
R03; Pemegang kartu dapat melihat saldo tabungannya saat ini.; 0
NR01; Sistem melakukan backup data minimal setiap satu minggu sekali.; 0
2. Memberi label pada setiap data set. Proses pelabelan ini melibatkan ahli dalam bidang rekayasa perangkat lunak. Untuk setiap pernyataan dalam sebuah data set, ahli akan diminta untuk memberikan label 0 jika pernyataan kebutuhan tersebut bukan noise, dan label 1 pernyataan kebutuhan tersebut noise. Ada tiga orang ahli yang akan dilibatkan dalam pelabelan pernyataan kebutuhan ini. Dari setiap data set, maka akan dibentuk data set dengan dua kesepakatan dan data set dengan tiga kesepakatan. Data set dengan dua kesepakatan adalah data set dengan label akhir merupakan label mayoritas diantara label yang diberikan oleh ketiga ahli. Data set dengan tiga kesepakatan adalah data set dengan aturan sebagai berikut: (a) jika salah satu ahli memberi label 0, maka label akhir dari pernyataan kebutuhan tersebut adalah 0, dan (b) jika semua ahli memberi label 1, maka label akhir dari pernyataan kebutuhan tersebut adalah 1.

3. Melakukan klasterisasi menggunakan metode yang diusulkan terhadap data set yang disediakan. Proses klasterisasi ini akan dilakukan sebanyak 10 kali pada setiap data set, dengan nilai ambang batas (threshold) kemiripan kata dari 0.0 sampai dengan 0.9. Dari hasil tersebut akan dihitung nilai kesepakatan dengan hasil dari para ahli. Perhitungan kesepakatan ini akan menggunakan kappa (Cohen, 1960). Dari hasil pengujian ini, akan ditentukan nilai ambang batas paling optimal (t_o) untuk menghasilkan kesepakatan terbaik. Dari hasil ini, maka akan diuji kembali sebanyak 10 kali dengan nilai ambang batas gradual mulai dari $t_o - 0.05$ sampai dengan $t_o + 0.05$. Hasil ini kemudian diolah lagi untuk menentukan nilai ambang batas paling optimal (t_o) terbaru. Nilai ini akan menjadi nilai ambang batas kemiripan kata terbaik untuk pendeteksian noise menggunakan metode yang diusulkan.

BAB 4

HASIL DAN ANALISIS

4.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari beberapa pernyataan kebutuhan. Data yang akan digunakan dalam pengujian berjumlah ± 628 pernyataan kebutuhan perangkat lunak yang telah diekstraksi secara manual dari 14 dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Setiap kebutuhan akan diberikan label oleh tiga penilai atau ekspert. Dataset ini selanjutnya akan disebut sebagai Dataset Asli (DA). Label pada dataset asli merupakan indikasi derau pada sebuah kebutuhan perangkat lunak. Masing-masing penilai dapat memberikan label berupa nilai *boolean* 1 (*true*) atau 0 (*false*). Label 1 menandakan bahwa kebutuhan tersebut merupakan *noise* atau derau pada sebuah dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) dan label 0 menandakan bahwa pernyataan kebutuhan tersebut bukan *noise* atau derau. Berikut adalah beberapa dataset yang digunakan pada penelitian ini.

Tabel 4.1 Dataset Penelitian

No.	ID	Nama Sistem	Jumlah
1.	DA-1	<i>Submit Job</i>	13
2.	DA-2	<i>System Administrator</i>	17
3.	DA-3	<i>Archive Administrator</i>	39
4.	DA_4	<i>SPG Application</i>	6
5.	DA-5	<i>System Administrator Staf</i>	33
6.	DA-6	<i>Software Development</i>	65
7.	DA-7	<i>Display System</i>	106
8.	DA-8	<i>Internet Acces</i>	64
9.	DA-9	<i>Meeting Initiator</i>	13
10.	DA-10	<i>Online System</i>	17
11.	DA-11	<i>Library System</i>	86
12.	DA-12	<i>IMSETY System</i>	70
13.	DA-13	<i>Manage Student Information</i>	24
14.	DA-14	<i>PHP Project</i>	75

4.2 Proses Dataset Awal

Dari dataset yang telah dikumpulkan langkah selanjutnya melabeli masing-masing kebutuhan oleh ekspert. Penelitian ini menggunakan tiga orang ekspert. Masing-masing ekspert akan memberikan label berupa nilai *boolean* 1 (*true*) atau 0 (*false*). Di mana 1 menandakan kebutuhan tersebut *noise* atau derau dan 0 bukan *noise* atau derau. Tabel 4.2 merupakan contoh dataset asli yang sudah dilabeli oleh expert. Data yang akan digunakan dalam pengujian adalah data derau mayoritas.

Tabel 4.2 Contoh Dataset Asli

ID Dataset Asli	Pernyataan Kebutuhan	Derau exp 1	Derau exp 2	Derau exp 3	Derau mayoritas
1	submit jobs with the associated deadline, cost, and execution time	0	0	0	0
2	query the cluster to establish the current cost per unit time for submitting new jobs	0	0	0	0
3	monitor the status of submitted jobs	0	0	0	0
4	cancel jobs submitted by him	0	0	0	0
5	check his credit balance	1	1	1	1
6	check his usage history	0	0	0	0
7	check the status of each node of the cluster	0	0	0	0
8	check the usage pattern history of the cluster	0	0	0	0
9	check the status of all submitted jobs	0	0	0	0
10	check the load on each node of the cluster	0	0	0	0
11	alter the cost structure of the cluster	0	0	0	0
12	alter the scheduling policy of the cluster	0	0	0	0
13	cancel, suspend, and resume any job	0	0	0	0

4.3 Skenario Pengujian

Terdapat 2 skenario pengujian untuk mengevaluasi metode usulan. Skenario 1 usulan metode akan diimplementasi pada dataset asli menggunakan 2 cara penentuan klaster derau yaitu :

1. Klaster derau atau noise ditentukan berdasarkan rata-rata jarak data ke titik pusat klaster terdekat atau metode A.
2. Klaster derau atau noise ditentukan berdasarkan rata-rata jarak data ke titik pusat klaster terjauh atau metode B.

Pada skenario 2 usulan metode akan diimplementasi pada dataset sintetis menggunakan 2 cara penentuan klaster derau yaitu :

1. Klaster derau ditentukan berdasarkan rata-rata jarak data ke titik pusat klaster terdekat (metode A).
2. Klaster derau ditentukan berdasarkan rata-rata jarak data ke titik pusat klaster terjauh (metode B).

Adapun dataset sintetis dibangun berdasarkan salah satu dataset asli yang telah disaring sehingga tidak mengandung pernyataan kebutuhan yang terklasifikasi derau atau *noise*. Kemudian dataset ini ditambahkan dengan dataset lain dengan perbandingan sekitar 9:1, sehingga dataset sintetis yang baru mengandung 90% pernyataan dari dataset utama dan 10% pernyataan dari dataset campuran. Pernyataan dari dataset utama dilabeli dengan 0, yaitu tidak derau. Tabel 4.3. menunjukkan daftar dataset sintetis yang dibangun.

Tabel 4.3 Daftar Dataset Sintetis

No	ID	Dataset Utama		ID Dataset Campuran	Jumlah
		ID dan Nama	Jumlah		
1	DS-1	<i>Submit Job</i>	13	DA-2 & DA-3	2
2	DS-2	<i>System Administrator</i>	17	DA-1 & DA-3	2
3	DS-3	<i>Archive Administrator</i>	20	DA-1 & DA-4	2
4	DS-4	<i>SPG Application</i>	6	DA-3 & DA-2	2
5	DS-5	<i>System Administrator Staf</i>	20	DA-4 & DA-6	2
6	DS-6	<i>Software Development</i>	19	DA-5 & DA-4	2
7	DS-7	<i>Display System</i>	29	DA-8 & DA-5	2
8	DS-8	<i>Internet Acces</i>	25	DA-7 & DA-3	2
9	DS-9	<i>Meeting Initiator</i>	18	DA-8 & DA-7	2
10	DS-10	<i>Online System</i>	17	DA-1 & DA-9	2
11	DS-11	<i>Library System</i>	15	DA-13 & DA-8	2
12	DS-12	<i>IMSETY System</i>	51	DA-13 & DA-14	2

No	ID	Dataset Utama		ID Dataset Campuran	Jumlah
		ID dan Nama	Jumlah		
13	DS-13	<i>Manage Student Information</i>	17	DA-10 & DA-14	2
14	DS-14	<i>PHP Project</i>	24	DA-13 & DA-5	2

Pernyataan dari dataset campuran dilabeli dengan 1, yaitu derau. Contoh dataset sintetis dapat dilihat pada tabel 4.4 Seperti pada Tabel 4.4, jika dataset sintetis mengandung lebih dari 10, maka jumlah pernyataan kebutuhan derau berasal dari dataset campuran adalah >2 . Untuk dataset sintetis yang mengandung lebih dari satu derau, maka pernyataan kebutuhan yang terklasifikasi derau diambil dari dataset campuran yang berbeda

Tabel 4.4 Contoh Dataset Sintetis

No	ID Dataset Campuran	Pernyataan Kebutuhan	ID Dataset Asli	Derau
1	DS-1	The users have sufficient knowledge of computers.	DA-1	0
2	DS-1	The University computer should have Internet connection and Internet server capabilities.	DA-1	0
3	DS-1	The users know the English language, as the user interface will be provided in English	DA-1	0
4	DS-1	The product can access the university student database	DA-1	0
5	DS-1	The system shall provide the users with logon capabilities.	DA-1	0
6	DS-1	The Online Library System is also supported on mobile devices such as cell phones.	DA-1	0
7	DS-1	The system can alert the Librarian or the administrator in case of any problems.	DA-1	0
8	DS-1	The system shall allow the users to access the system from the Internet using HTML or it's derivative technologies.	DA-1	0
9	DS-1	The system uses a web browser as an interface.	DA-1	0
10	DS-1	Since all users are familiar with the general usage of browsers, no specific training is required.	DA-1	0
11	DS-1	The system is user friendly and self-explanatory.	DA-1	0

No	ID Dataset Campuran	Pernyataan Kebutuhan	ID Dataset Asli	Derau
12	DS-1	All staff can view the details of any student	DA-2	1
13	DS-1	All The above users have similar rights.	DA-10	1

4.4 Hasil Uji Skenario

Pada tahap pengujian skenario meliputi persiapan data, evaluasi hasil prediksi pendeteksi noise atau derau menggunakan lima (5) pengukuran evaluasi yaitu *kappa*, Gwet's Ac1, *Sensitivitas*, *Spesivitas*, *F1 Score*. Beberapa skenario pengujian juga akan dilakukan untuk memperdalam evaluasi pendeteksi *noise* atau derau di dalam kebutuhan perangkat lunak.

4.4.1 Hasil Uji Skenario 1

Uji coba pada skenario satu dilakukan dengan mengimplementasi usulan metode pada dataset asli. Hasil dari skenario satu dapat dilihat pada tabel 4.5. Pada tabel hasil uji coba skenario 1 dapat dilihat bahwa penentuan klaster derau atau *noise* menggunakan rata-rata jarak *centroid* terdekat (metode A) dan rata-rata *centroid* terjauh (metode B). Terlihat juga pada tabel tersebut jumlah klaster hampir semuanya jumlah klaster optimal adalah satu. Hal tersebut disebabkan karena datanya belum tercampur sehingga kalimat pernyataan kebutuhannya tidak bervariasi.

Tabel 4.5 Hasil Uji Skenario 1

Dataset	Jumlah Kebutuhan	Jumlah Klaster	Kappa Score		Gwet's ACI		Sensitivitas		Spesivitas		F1-Score	
			Penentua n Klaster Derau Cara A	Penentu an Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penent uan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentua n Klaster Derau Cara A	Penentua n Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B
DA-1	13	2	-0.13	0.05	0.58	-0.35	0.00	1.00	0.75	0.25	0.00	0.18
DA-2	17	2	-0.10	0.03	0.70	0.51	0.00	1.00	0.25	0.19	0.00	0.13
DA-3	39	2	0.07	-0.05	0.41	-0.07	1.00	0.00	0.60	0.3z9	0.11	0.00
DA-4	6	2	-0.29	0.18	0.02	0.03	0.00	1.00	0.06	0.04	0.00	0.04
DA-5	33	2	0.00	0.00	0.93	-0.87	0.00	0.00	0.93	0.06	0.00	0.00
DA-6	65	2	-0.16	0.06	0.33	-0.15	0.04	0.95	0.81	0.14	0.06	0.51
DA-7	106	2	-0.23	0.12	0.36	-0.10	0.00	1.00	0.65	0.35	0.00	0.32
DA-8	64	2	-0.11	0.02	0.75	-0.59	0.00	1.00	0.91	0.09	0.00	0.24
DA-9	33	2	-0.04	0.00	0.90	-0.82	0.00	1.00	0.93	0.06	0.00	0.06
DA-10	17	2	0.01	-0.01	0.19	0.06	0.05	0.05	0.53	0.47	0.02	0.18
DA-11	86	2	-0.04	0.02	0.53	-0.29	0.25	0.78	0.71	0.29	0.13	0.20
DA-12	70	2	0.00	0.00	0.79	-0.61	0.00	0.00	0.82	0.17	0.00	0.00
DA-13	24	2	-0.24	0.19	0.10	0.16	0.00	0.00	0.52	0.48	0.00	0.35
DA-14	75	2	0.00	0.00	0.95	-1.00	0.00	0.00	0.96	0.00	0.00	0.00
Average			-0.09	0.04	0.54	-0.29	0.03	0.22	0.73	0.21	0.02	0.16

Salah satu contoh hasil deteksi *noise* atau derau uji coba skenario 1 dapat di lihat pada tabel 4.6 dan tabel 4.7. Pada tabel tersebut klaster derau ditentukan berdasarkan rata-rata jarak data ke titik pusat klaster terjauh dan jarak rata-rata ke titik klaster terdekat. Terlihat juga klaster *noise* atau deraunya 1 maka semua kebutuhan dianggap derau.

Tabel 4.6 Contoh Derau Hasil Uji Skenario 1 (Rata-rata hasil jarak terbesar)

Nama File	Requirements Statement	Clust Label	Distance	Noise Cluster	Prediction	Manual
DA-2	The system administrator must be able to deactivate and reactivate student account login	0	0.25	[0]	1	0
DA-2	The system administrator must be able to force the sending of a new password to a student via email	0	0.24	[0]	1	0
DA-2	The system administrator must be able to change any of a student's details	0	0.21	[0]	1	0
DA-2	Student can be register on the system and fill in all detail and forward to choose project/supervisor	0	0.29	[0]	1	0
DA-2	Student can change detail if information is incorrect such as telephone number	0	0.25	[0]	1	0
DA-2	Student can change his login password at any time for security reason	0	0.27	[0]	1	0
DA-2	Student can request his password if he/she forgotten the	0	0.28	[0]	1	0

Nama File	Requirements Statement	Clust Label	Distance	Noise Cluster	Prediction	Manual
	password					
DA-2	All staff can view the details of any student	0	0.26	[0]	1	0
DA-2	Certain staff may be designated as Unit or Cohort Co-ordinators and can change the details of any student doing their unit or project cohort	0	0.23	[0]	1	0
DA-2	Unit Cohort co-ordinator can change student detail for incorrect information	0	0.19	[0]	1	0
DA-2	Unit Cohort co-ordinator can view student information and monitor their progress	0	0.22	[0]	1	0
DA-2	Unit Cohort co-ordinator can list all students in different period of different group	1	0.02	[0]	0	0
DA-2	Unit Cohort co-ordinator can reset the student's password if required	0	0.20	[0]	1	0
DA-2	System Administrator can list all students in different period of different group to check any error	1	0.04	[0]	0	0
DA-2	System Administrator can reset the student's password if	0	0.22	[0]	1	0

Nama File	Requirements Statement	Clust Label	Distance	Noise Cluster	Prediction	Manual
	required					
DA-2	Administration Staff can list all students in different period of different group	1	0.02	[0]	0	0
DA-2	Administration Staff can make a school announcement	0	0.33	[0]	1	1
Kappa						0.03
Gwet's Acl						-0.51
Sensivitas						1
Spesivitas						0.19
F1 Score						0.13

Tabel 4.7 Contoh Deteksi Derau Hasil Uji Skenario 1 (Rata-rata terkecil)

Nama File	Requirements Statement	Clust Label	Distance	Noise Cluster	Prediction	Manual
DA-2	The system administrator must be able to deactivate and reactivate student account login	0	0.25	[1]	0	0
DA-2	The system administrator must be able to force the sending of a new password to a student via email	0	0.24	[1]	0	0
DA-2	The system administrator must be able to change any of a student's details	0	0.21	[1]	0	0
DA-2	Student can be register on the system and fill in all detail and forward to choose project/supervisor	0	0.29	[1]	0	0
DA-2	Student can change	0	0.25	[1]	0	0

Nama File	Requirements Statement	Clust Label	Distance	Noise Cluster	Prediction	Manual
	detail if information is incorrect such as telephone number					
DA-2	Student can change his login password at any time for security reason	0	0.	[1]	0	0
DA-2	Student can request his password if he/she forgotten the password	0	0.28	[1]	0	0
DA-2	All staff can view the details of any student	0	0.26	[1]	0	0
DA-2	Certain staff may be designated as Unit or Cohort Co-ordinators and can change the details of any student doing their unit or project cohort	0	0.23	[1]	0	0
DA-2	Unit Cohort co-ordinator can change student detail for incorrect information	0	0.19	[1]	0	0
DA-2	Unit Cohort co-ordinator can view student information and monitor their progress	0	0.22	[1]	0	0
DA-2	Unit Cohort co-ordinator can list all students in different period of different group	1	0.02	[1]	1	0
DA-2	Unit Cohort co-ordinator can reset the student password if required	0	0.20	[1]	0	0
Kappa						-0.10
Gwet's Ac1						0.70
Sensivitas						0
Spesivitas						0.25
F1 Score						0

4.4.2 Hasil Uji Skenario 2

Pada uji coba skenario 2 dilakukan dengan mengimplementasikan metode usulan pada dataset sintetis atau mencampurkan satu dokumen dengan dokumen yang lain. Hasil dari skenario 2 dapat dilihat pada tabel 4.8. Pada tabel hasil uji coba skenario 2 tersebut dapat dilihat bahwa penentuan kluster derau atau *noise* menggunakan rata-rata jarak *centroid* terdekat (metode A) dan rata-rata jarak *centroid* terjauh (metode B) hasilnya juga bervariasi.

Tabel 4.8 Hasil Uji Coba Skenario 2

Dataset	Jumlah Kebutuhan	Jumlah Klaster	Kappa Score		Gwet's AC1		Sensitivitas		Spesivitas		F1-Score	
			Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B
DS-1	15	2	-0.19	0.07	0.54	-0.33	0.00	1.00	0.77	0.23	0.00	0.29
DS-2	19	2	-0.14	0.04	0.66	-0.47	0.00	1.00	0.82	0.18	0.00	0.22
DS-3	22	2	-0.16	0.07	0.48	-0.23	0.00	1.00	0.70	0.30	0.00	0.22
DS-4	8	2	-0.56	0.71	-0.72	0.78	0.00	1.00	0.17	0.83	0.00	0.08
DS-5	22	2	-0.10	0.02	0.78	-0.64	0.00	1.00	0.90	0.10	0.00	0.81
DS-6	21	2	-0.16	0.06	0.54	-0.31	0.00	1.00	0.74	0.26	0.00	0.22
DS-7	31	2	-0.12	0.07	0.37	-0.07	0.00	1.00	0.62	0.38	0.00	0.81
DS-8	27	2	-0.12	0.04	0.67	-0.46	0.00	1.00	0.80	0.20	0.00	0.17
DS-9	20	2	-0.15	0.05	0.60	-0.39	0.00	1.00	0.78	0.22	0.00	0.22
DS-10	19	2	-0.18	0.08	0.47	-0.23	0.00	1.00	0.71	0.29	0.00	0.25
DS-11	17	2	-0.23	0.21	-0.01	0.27	0.00	1.00	0.47	0.53	0.00	0.36
DS-12	53	2	-0.07	0.03	0.53	-0.23	0.00	1.00	0.69	0.31	0.00	0.10

Dataset	Jumlah Kebutuhan	Jumlah Klaster	Kappa Score		Gwet's AC1		Sensitivitas		Spesivitas		F1-Score	
			Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B	Penentuan Klaster Derau Cara A	Penentuan Klaster Derau Cara B
DS-13	19	2	-0.16	0.06	0.57	-0.35	0.00	1.00	0.76	0.24	0.00	0.24
DS-14	26	2	-0.10	0.02	0.77	-0.61	0.00	1.00	0.88	0.13	0.00	0.16
Average	-0.17	0.11	0.45	-0.23	0.00	1.00	0.00	1.00	0.70	0.30	0.00	0.30

Pada Tabel 4.9 dan Tabel 4.10 adalah salah contoh deteksi derau atau *noise* hasil uji coba skenario 2 klaster derau ditentukan berdasarkan rata-rata jarak data ke titik pusat klaster terkecil dan terbesar. Terlihat bahwa klaster yang *noise* atau derau dapat dikelompokkan ke dalam klaster yang sama berdasarkan konteks menggunakan klasterisasi *spectral*.

Pada Tabel 4.8 dilihat bahwa klaster yang noise atau derau dapat dikelompokkan ke dalam klaster yang sama tetapi cuma terdapat 2 klaster karena pada kalimat kebutuhan tersebut banyak mengulang kata yang sama. Walaupun presisi rendah namun sistem masih bisa menghasilkan *recall* yang tinggi.

Tabel 4.9 Contoh Hasil Deteksi Derau Skenario 2 (rata-rata terbesar)

Nama File	Requirements Statemen	Clust Label	Distance	Noise Cluster	Prediction	Manual
DS-4	User can see detailed information about SGP	0	0.25	[1]	0	0
DS-4	User can apply for SGP	0	0.22	[1]	0	0
DS-4	User can application update form for SGP	0	0.17	[1]	0	0
DS-4	User can update personal data	0	0.26	[1]	0	0
DS-4	User can upload application related documents	0	0.28	[1]	0	0
DS-4	User can perform psychological test	1	0.24	[1]	1	0
DS-4	The archive administrator can create a new client space.	1	0.25	[1]	1	1
DS-4	Student can request his password if he/she forgotten the password	1	0.25	[1]	1	1
Kappa						0.71
Gwet's Acl						0.78
Sensivitas						1.00
Spesivitas						0.83
F1 Score						0.08

Tabel 4.10 Contoh Hasil Deteksi Derau Skenario 2 (rata-rata terkecil)

Nama File	Requirements Statemen	Clust Label	Distance	Noise Cluster	Prediction	Manual
DS-4	User can see detailed information about SGP	0	0.25	[0]	1	0
DS-4	User can apply for SGP	0	0.22	[0]	1	0
DS-4	User can application update form for SGP	0	0.17	[0]	1	0
DS-4	User can update personal data	0	0.26	[0]	1	0
DS-4	User can upload application related documents	0	0.28	[0]	1	0

Nama File	Requirements Statemen	Clust Label	Distance	Noise Cluster	Prediction	Manual
DS-4	User can perform psychological test	1	0.24	[0]	0	0
DS-4	The archive administrator can create a new client space.	1	0.25	[0]	0	1
DS-4	Student can request his password if he/she forgotten the password	1	0.25	[0]	0	1
Kappa						-0.56
Gwet's Ac1						-0.72
Sensivitas						0.00
Spesivitas						0.17
F1 Score						0.00

4.5. Analisis Hasil Skenario 1 dan Skenario 2

Berdasarkan hasil pengujian (Tabel 4.5 dan 4.8) dapat dilihat bahwa kinerja metode masih rendah. Umumnya, pada dataset asli, tingkat sensitivitas dan spesivitas sangat rendah. Hal ini disebabkan dua hal. Pertama, distribusi nilai fitur yang dihasilkan dari preproses pernyataan kebutuhan yang dihasilkan tidak tersebar dengan baik. Gambar 4.1 menunjukkan matriks *Affinity* dari dataset DA-1 yang berisi pernyataan kebutuhan.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13
R1	1.00	0.96	0.85	0.83	0.93	0.89	0.82	0.89	0.92	0.82	0.73	0.79	0.78
R2	0.96	1.00	0.89	0.84	0.94	0.89	0.87	0.91	0.93	0.89	0.79	0.78	0.79
R3	0.85	0.89	1.00	0.91	0.91	0.82	0.93	0.84	0.97	0.91	0.74	0.67	0.82
R4	0.83	0.84	0.91	1.00	0.87	0.73	0.76	0.74	0.90	0.73	0.60	0.64	0.87
R5	0.93	0.94	0.91	0.87	1.00	0.91	0.88	0.91	0.95	0.87	0.78	0.77	0.86
R6	0.89	0.89	0.82	0.73	0.91	1.00	0.86	0.98	0.91	0.85	0.71	0.85	0.75
R7	0.82	0.87	0.93	0.76	0.88	0.86	1.00	0.89	0.93	0.96	0.86	0.71	0.76
R8	0.89	0.91	0.84	0.74	0.91	0.98	0.89	1.00	0.91	0.89	0.77	0.88	0.77
R9	0.92	0.93	0.97	0.90	0.95	0.91	0.93	0.91	1.00	0.88	0.74	0.75	0.84
R10	0.82	0.89	0.91	0.73	0.87	0.85	0.96	0.89	0.88	1.00	0.86	0.69	0.74
R11	0.73	0.79	0.74	0.60	0.78	0.71	0.86	0.77	0.74	0.86	1.00	0.70	0.62
R12	0.79	0.78	0.67	0.64	0.77	0.85	0.71	0.88	0.75	0.69	0.70	1.00	0.70
R13	0.78	0.79	0.82	0.87	0.86	0.75	0.76	0.77	0.84	0.74	0.62	0.70	1.00

Gambar 4.1 Matriks Affinity dari dataset DA-1.

Terlihat bahwa perhitungan kemiripan kalimat menggunakan *cosine similarity* berdasarkan frekuensi kemunculan kata dalam setiap pernyataan kebutuhan menghasilkan nilai yang tidak terdistribusi dengan baik. Hal ini disebabkan dalam proses pembentukan matriks S semua jenis kata dipertimbangkan (seperti ditunjukkan dalam Tabel 3.1). Hal ini pada akhirnya menyebabkan banyaknya *false positif* dan *false negative*. Padahal di dalam kebutuhan ada dua notasi yang utama, yaitu aktor dan aksi yang dilakukan aktor. Aktor sendiri biasanya berbentuk kata benda. Aksi terdiri dari kata kerja dan kata benda yang menjadi obyek penderita dari aksi tersebut. Penyebab kedua adalah penentuan pernyataan derau didasarkan pada klaster dengan jarak paling dekat/jauh. Padahal dimungkinkan ada lebih dari satu pernyataan derau dalam satu dataset yang berada pada klaster yang berbeda. Untuk itu, dua perubahan pada metode deteksi derau dilakukan untuk menyelesaikan masalah di atas. Pertama, preproses diperbaiki dengan melakukan *stopword removal* pada jenis kata selain kata benda dan kata kerja. Selain itu, proses pemulihan kata benda bentuk tunggal dan kata kerja bentuk pertama juga diperbaiki. Kedua, ada suatu nilai ambang digunakan untuk menentukan apakah suatu pernyataan kebutuhan dengan nilai jarak terhadap tetangga klasternya merupakan derau atau bukan. Jika nilai jaraknya di atas nilai ambang, maka pernyataan tersebut diklasifikasikan sebagai derau.

Tabel 4.11 Nilai Reliabilitas untuk Dataset Asli

Dataset	Threshold									
	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
DA-1	0.20	0.52	0.67	0.71	0.92	0.92				
DA-2	0.27	0.52	0.74	0.67	0.70	0.79				
DA-3	0.65	0.85	0.92	0.92	0.92	0.92				
DA-4	0.40	0.40	0.73	0.73	0.80	0.80				
DA-5	-0.37	0.16	0.71	0.86	0.86	0.86				
DA-6	0.02	0.34	0.52	0.46	0.50	0.58				
DA-7	0.13	0.62	0.77	0.90	0.87	0.92				
DA-8	-0.12	0.32	0.70	0.80	0.84	0.86				
DA-9	-0.03	0.45	0.70	0.79	0.79	0.84				
DA-10	0.40	0.83	0.83	0.83	0.93	0.87				
DA-11	-0.14	0.47	0.70	0.78	0.79	0.79				
DA-12	0.13	0.63	0.82	0.84	0.92	0.92				
DA-13	0.06	0.60	0.83	0.95	0.95	0.90				

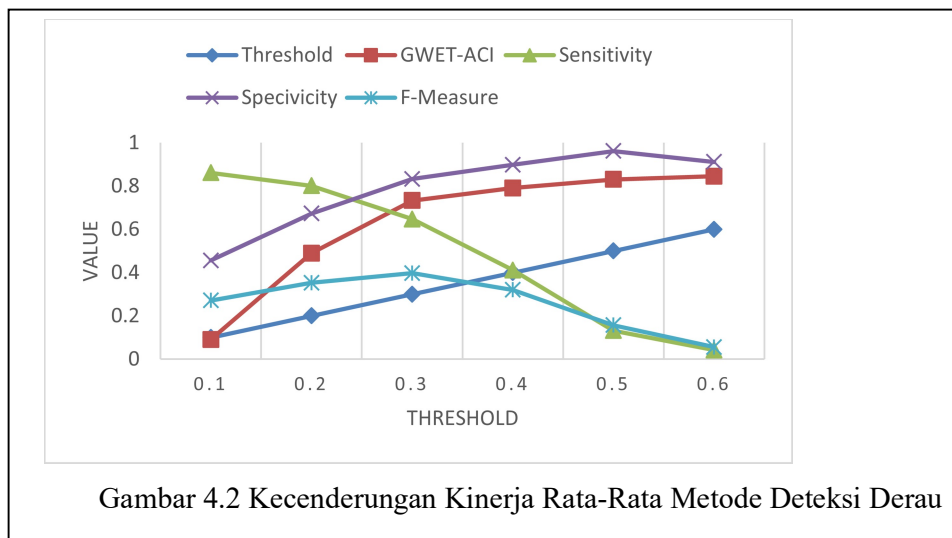
DA-14	-0.32	0.15	0.62	0.84	0.84	0.86	
Rata-rata	0.09	0.49	0.73	0.79	0.83	0.85	

Tabel 4.10 menunjukkan nilai reliabilitas berdasarkan perhitungan GWET-AC1 untuk dataset asli. Dapat kita lihat bahwa semakin naik nilai ambang batas, maka nilai reliability cenderung naik. Akan tetapi nilai reliabilitas yang tinggi tidak sepenuhnya menunjukkan bahwa metode deteksi ini sudah baik. Tabel 4.11 menunjukkan nilai rata-rata reliabilitas, sensitivitas, spesivitas, dan f-measure dari metode deteksi derau berbasis *spectral clustering* yang telah diperbaiki.

Tabel 4.12 Kinerja Rata-Rata Metode Deteksi Derau

Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
GWET-ACI	0.09	0.49	0.73	0.79	0.83	0.85				
Sensitivity	0.86	0.80	0.65	0.41	0.13	0.04				
Specivicity	0.46	0.67	0.83	0.90	0.96	0.91				
F-Measure	0.27	0.35	0.40	0.32	0.16	0.06				

Dapat dilihat bahwa kinerja rata-rata terbaik akan tercapai pada sekitaran ambang batas 0.3, dimana f-measure bernilai 0.4. Gambar 4.2 menunjukkan visualisasi kecenderungan hasil pengujian. Dari kecenderungan tersebut dapat diduga bahwa nilai optimal dari metode terdapat di antara nilai ambang batas 0.2 sampai dengan 0.4



Tabel 4.12 menunjukkan nilai rata-rata GWET-AC1, sensitivitas, spesivitas, dan f-measure untuk nilai ambang di antara 0.2 sampai dengan 0.4. Dapat dilihat bahwa kinerja terbaik dicapai pada nilai ambang 0.35. Pada nilai ambang tersebut metode deteksi derau dapat memiliki nilai reliabilitas 0.788, sensitivitas 0.636, dan sensitivitas 0.874. Nilai reliabilitas 0.788 menunjukkan bahwa metode ini tergolong substansial, artinya metode dapat bekerja baik seperti anotator ahli.

Tabel 4.13 Kinerja Rata-Rata Metode Deteksi Derau

Threshold	0.2	0.25	0.3	0.32	0.35	0.37	0.4
GWET-ACI	0.490	0.625	0.733	0.771	0.788	0.784	0.791
Sensitivity	0.801	0.774	0.648	0.648	0.636	0.506	0.412
Specivicity	0.673	0.757	0.833	0.860	0.874	0.881	0.898
F-Measure	0.353	0.392	0.397	0.429	0.438	0.377	0.320

Tabel 4.13 juga menunjukkan bahwa sekitar 35% pernyataan kebutuhan derau (menurut ahli) dinyatakan bukan derau oleh metode. Nilai sensitivitas dari metode deteksi derau menunjukkan bahwa ada sekitar 35% pernyataan kebutuhan yang salah klasifikasi seperti itu. Pada umumnya, dataset dengan nilai sensitivitas rendah memiliki jumlah pernyataan derau yang cukup banyak, yaitu 9 s.d. 20 pernyataan (Tabel 4.14).

Tabel 4.14 Kinerja Rata-Rata Metode Deteksi Derau

	Derau	Pernyataan	Rasio	Sensitivitas
DA-1	1	13	0.08	1.00
DA-2	3	17	0.18	0.66
DA-3	1	39	0.03	0.00
DA-4	1	33	0.03	1.00
DA-5	3	33	0.09	0.66
DA-6	20	65	0.31	0.35
DA-7	4	106	0.04	1.00
DA-8	6	64	0.09	0.50
DA-9	18	285	0.06	0.00
DA-10	2	17	0.12	1.00
DA-11	14	86	0.16	0.50
DA-12	4	70	0.06	1.00
DA-13	3	24	0.13	0.66
DA-14	9	75	0.12	0.55

Dari pernyataan kebutuhan derau yang tidak berhasil diidentifikasi, terdapat pernyataan yang sebenarnya memang bukan merupakan pernyataan kebutuhan yang tidak relevan terhadap sistem, akan tetapi merupakan pernyataan non kebutuhan. Pernyataan non kebutuhan ini diantaranya terkait aspek manajemen

proyek, perancangan, implementasi, dan pengujian. Sebagai contoh, pernyataan kebutuhan R2 dari dataset DA-8 dan R dari dataset DA-y.

“The Twitter client is a two months project that cost us almost seventy million dollars.” (R2, dataset DA-8)

“This software is being developed to for customer Xuan Gu an employee for the CISE Department at the University of Florida.” (R1, dataset DA-6)

Tabel 4.13 juga menunjukkan bahwa metode sudah baik dalam mengklasifikasi pernyataan kebutuhan yang bukan derau. Hal ini ditunjukkan dengan nilai spesivisitas 0.874 pada nilai ambang 0.35. Beberapa pernyataan yang salah klasifikasi oleh metode (dianggap sebagai derau) seringkali terdiri dari beberapa kalimat. Sebagai contoh, pernyataan R4 dari dataset DA-5 dan R13 dari dataset DA-14 di bawah ini:

“System provides a user-friendly interface for designing queries against all record types. Staff can select fields to query select values from picklist of possible values select regular expressions from drop-down menu, and use a full range of Boolean operators. Administrators control staff access to tables and fields.” (R4, dataset DA-5)

“During typing the PHP inspector is run and highlighting of errors (a squiggly line under the code) occurs for the file. Errors detected during typing are not displayed in the problems view. The build options in the project menu determine when the eclipse builder is run. The eclipse builder is potentially a long running operation. Depending on the changes made, every file in the workspace could be rebuilt. Eclipse attempts to determine the files that have changed based on the project dependencies. The eclipse builder is run as a background task so that the developer can continue to work. Problems detected during the build phase will show up in the problems view and will be highlighted using a squiggly line under the code.” (R13, dataset DA-14)

Hal ini menyebabkan pernyataan kebutuhan seperti ini berada di tengah-tengah banyak pernyataan kebutuhan yang lain. Sehingga seringkali dianggap sebagai derau.

BAB 5

PENUTUP

5.1 Kesimpulan

Evaluasi pendeteksian *noise* atau derau di dalam kebutuhan perangkat lunak memiliki manfaat di dalam pengembangan perangkat lunak. Manfaat dari deteksi *noise* atau derau di dalam kebutuhan perangkat lunak diantaranya menghindari kesalahan sedini mungkin sehingga dapat mengurangi biaya perbaikan dan dapat memberikan informasi tentang prioritas kebutuhan perangkat lunak. Metode deteksi derau yang dibangun dapat bekerja dengan baik. Metode ini memiliki tingkat reliabilitas substantial (nilai GWET AC1 0.788) dan f-measure 0.438 pada nilai ambang 0.35 untuk dataset asli (DA).

Preproses pada metode deteksi derau yang dibangun akan mengekstaksi hanya kata benda (dalam bentuk tunggal) dan kata kerja (dalam bentuk pertama) saja dari setiap pernyataan. Sebuah pernyataan kebutuhan dinyatakan sebagai derau jika nilai jarak pernyataan terhadap kluster tetangganya lebih tinggi daripada nilai ambang batas yang telah ditentukan dalam penelitian (yaitu 0.35).

5.2 Saran

Berdasarkan hasil pengujian, metode deteksi derau yang dibangun memiliki nilai sensitivitas 0.636. Artinya ada sekitar 35% pernyataan yang menurut ahli merupakan pernyataan kebutuhan derau akan tetapi tidak teridentifikasi oleh metode. Kedepan penelitian dapat diarahkan untuk meningkatkan sensitivitas metode dengan menambahkan metode untuk mengklasifikasi setiap pernyataan sebagai pernyataan kebutuhan atau non-kebutuhan. Selain itu, penelitian kedepan juga dapat diarahkan untuk mengembangkan fungsi jarak yang lebih baik yang dapat menghasilkan hasil klaster yang lebih akurat.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Cai, X. *et al.* (2011) ‘Simultaneous Clustering and Noise Detection for Theme-based Summarization’, *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 491–499. Available at: <http://aclweb.org/anthology/I11-1055>.
- Cohen, J. (1960) ‘A Coefficient of Agreement for Nominal Scales’, *Education Psychology Measurement*, 3.
- Garcia, © E and Co, A. (2015) ‘Cosine Similarity Tutorial’, pp. 4–10.
- Gupta, H. and Srivastava, R. (2014) ‘K-means Based Document Clustering with Automatic “ K ” Selection and Cluster Refinement’, *International Journal of Computer Science and Mobile Applications*, 2, pp. 7–13.
- Knijff, J. De, Frasinicar, F. and Hogenboom, F. (2013) ‘Data & Knowledge Engineering Domain taxonomy learning from text: The subsumption method versus hierarchical clustering’, *Data & Knowledge Engineering*, 83(0), pp. 54–69. doi: 10.1016/j.datak.2012.10.002.
- Luxburg, U. Von (2007) ‘A Tutorial on Spectral Clustering’, pp. 1–32.
- Meyer, B. (1985) ‘on Formalism in Specifications.’, *IEEE Software*, 2(1), pp. 6–26. doi: 10.1109/MS.1985.229776.
- Montes-y-gómez, M., Gelbukh, A. F. and López-lópez, A. (2002) ‘Detecting Deviations in Text Collections: An Approach Using Conceptual Graphs’, *Mexican International Conference on Artificial Intelligence*, pp. 176–184. doi: 10.1007/3-540-46016-0_19.
- Popat, S. K. and Emmanuel, M. (2014) ‘Review and Comparative Study of Clustering Techniques’, *(IJCSIT) International Journal of Computer Science and Information Technologies*, 5(1), pp. 805–812.
- Prasetyo, E. (2014) *Data Mining: Mengolah Data Menjadi Informasi dengan Matlab*.
- Rossi, A. (1999) ‘Incentives in managerial compensation: a survey of experimental research’, *ROCK Working Papers*.
- Saiyad, N. Y., Prajapati, H. B. and Dabhi, V. K. (2016) ‘A Survey of Document Clustering using Semantic Approach’, *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 6(4), pp. 2555–2562. doi: 10.1109/ICTEE.2012.62.
- Siahaan, D. (2012) *Analisis Kebutuhan dalam Rekayasa Perangkat Lunak*.
- Sommerville (2007) *Software Enginnering, 8th Edition, Addison Wesley, Boston*.
- Uw, S. *et al.* (2001) ‘On spectral clustering: Analysis and an algorithm’, *Advances in Neural Information Processing Systems 14*, pp. 849–856. doi: 10.1.1.19.8100.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Patricia Gertrudis Manek, lahir di Kefamenanu pada tanggal 17 Maret 1986. Penulis merupakan anak kelima dari enam bersaudara. Penulis telah menempuh pendidikan mulai SDK Yaperna Oemanu Kefamenanu Sepanjang (1995-2000), SMP N 1 Kefamenanu (2000-2002), SMA N 1 Kefamenanu (2002-2004), menempuh Perguruan Tinggi S1 STMIK PPKIA PRADNYA PARAMITA Malang spanjang (2004-2008) dan terakhir sebagai mahasiswa S2 Teknik Informatika ITS Surabaya (2016-2019). Penulis yang mempunyai kegemaran nonton menyanyi, membaca dan traveling. Penulis dapat dihubungi melalui e-mail: patricia.manek@gmail.com.

(Halaman ini sengaja dikosongkan)